SAS/QC® 9.1

User's Guide

The Power to Know®

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2004. *SAS/QC® 9.1 User's Guide*. Cary, NC: SAS Institute Inc.

SAS/QC® 9.1 User's Guide

ISBN 1-59047-242-X

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, January 2004

SAS Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at **support.sas.com/pubs** or call 1-800-727-3228.

Contents

Part 1. The ANOM Procedure **1**

Chapter 1. PROC ANOM and General Statements . . . 3
Chapter 2. XCHART Statement . . . 13
Chapter 3. PCHART Statement . . . 53
Chapter 4. UCHART Statement . . . 83
Chapter 5. BOXCHART Statement . . . 109
Chapter 6. INSET Statement . . . 141
Chapter 7. References . . . 157

Part 2. The CAPABILITY Procedure **159**

Introduction . . . 161
Chapter 8. PROC CAPABILITY and General Statements . . . 163
Chapter 9. CDFPLOT Statement . . . 225
Chapter 10. COMPHISTOGRAM Statement . . . 245
Chapter 11. HISTOGRAM Statement . . . 277
Chapter 12. INSET Statement . . . 353
Chapter 13. INTERVALS Statement . . . 377
Chapter 14. OUTPUT Statement . . . 391
Chapter 15. PPPLOT Statement . . . 407
Chapter 16. PROBPLOT Statement . . . 429
Chapter 17. QQPLOT Statement . . . 461
References . . . 501

Part 3. The CUSUM Procedure **507**

Introduction . . . 509
Chapter 18. PROC CUSUM Statement . . . 511
Chapter 19. XCHART Statement . . . 519
Chapter 20. INSET Statement . . . 577

References 583

Part 4. The FACTEX Procedure **585**

Chapter 21. Introduction to the FACTEX Procedure 587

Chapter 22. Details of the FACTEX Procedure 599

Chapter 23. Theory of Orthogonal Designs 661

References 669

Part 5. The ISHIKAWA Procedure **671**

Chapter 24. Introduction to the ISHIKAWA Environment 673

Chapter 25. Details of the ISHIKAWA Environment 687

References 749

Part 6. The MACONTROL Procedure **751**

Introduction 753

Chapter 26. PROC MACONTROL Statement 755

Chapter 27. EWMACHART Statement 763

Chapter 28. MACHART Statement 819

Chapter 29. INSET Statement 863

References 869

Part 7. The OPTEX Procedure **871**

Chapter 30. Introduction to the OPTEX Procedure 873

Chapter 31. Details of the OPTEX Procedure 887

References 949

Part 8. The PARETO Procedure **951**

Introduction 953

Chapter 32. PROC PARETO Statement 955

Chapter 33. VBAR Statement 961

Chapter 34. HBAR Statement 997

Chapter 35. INSET Statement 1031

Chapter 36. Details and Examples 1047
References 1077

Part 9. The RELIABILITY Procedure **1079**

Chapter 37. The RELIABILITY Procedure 1081
References 1217

Part 10. The SHEWHART Procedure **1219**

Introduction 1221
Chapter 38. PROC SHEWHART and General Statements 1227
Chapter 39. BOXCHART Statement 1237
Chapter 40. CCHART Statement 1303
Chapter 41. IRCHART Statement 1345
Chapter 42. MCHART Statement 1389
Chapter 43. MRCHART Statement 1435
Chapter 44. NPCHART Statement 1481
Chapter 45. PCHART Statement 1525
Chapter 46. RCHART Statement 1571
Chapter 47. SCHART Statement 1611
Chapter 48. UCHART Statement 1649
Chapter 49. XCHART Statement 1689
Chapter 50. XRCHART Statement 1735
Chapter 51. XSCHART Statement 1787
Chapter 52. INSET and INSET2 Statements 1833
Chapter 53. Dictionary of Options 1851
Chapter 54. Graphical Enhancements 1927
Chapter 55. Tests for Special Causes 1975
Chapter 56. Specialized Control Charts 1999
Chapter 57. Interactive Control Charts 2039
References 2049

Part 11. Appendices **2055**

Appendix A. The GAGE Application 2057

Appendix B. The RELIABILITY Graphical Interface 2085

Appendix C. Functions 2089

Appendix D. Special Fonts in SAS/QC Software 2117

Appendix E. References 2123

Subject Index **2125**

Syntax Index **2149**

Acknowledgments

Credits

Documentation

Writing	Michael J. Cybrynski, Gordon Johnston, Julie LaBarr, Sharad S. Prabhu, Bucky Ransdell, Robert N. Rodriguez, Elizabeth Shamseldin, Randall D. Tobias
Editing	Virginia Clark, Sharad S. Prabhu, Robert N. Rodriguez, Donna Sawyer, Julie Simmons
Document Management and Production	Tim Arnold

Software

The procedures in SAS/QC software were implemented by the Statistical Quality Improvement Research and Development Department. Substantial support was given to the project by other members of the Analytical Solutions Division. The Core Development Division, Display Products Division, Graphics Division, and Host Systems Division also contributed to this product.

ANOM	Bucky Ransdell, Robert N. Rodriguez
CAPABILITY	Bucky Ransdell, Robert N. Rodriguez
CUSUM	Bucky Ransdell, Robert N. Rodriguez
FACTEX	Randall D. Tobias
ISHIKAWA	Michael J. Cybrynski
MACONTROL	Bucky Ransdell, Robert N. Rodriguez
OPTEX	Randall D. Tobias
PARETO	Bucky Ransdell, Robert N. Rodriguez
RELIABILITY	Gordon Johnston
SHEWHART	Michael J. Cybrynski, Bucky Ransdell, Robert N. Rodriguez
Testing	John Johnson, Jeanne Martin

Support Groups

Quality Assurance	Jack J. Berry, Brett Chapman, Jonathan Chapman, Patricia E. Mullins, Sumita Biswas, Audrey Ventura
Technical Support	David Schlotzhauer, Annette Sanders, Elizabeth Edwards

Acknowledgments

Many people have been instrumental in the development of SAS/QC software. The individuals acknowledged here have been especially helpful.

Melvin T. Alexander	Westinghouse Electric Corporation
Kevin Anderson	Motorola Inc.
Robert V. Baxley	Monsanto Company
Linda W. Blazek	Alcoa Laboratories
James L. Bossert	Eastman Kodak Company
Mike Boyko	Singer Link Flight Simulation Division
Bob Chiverton	G. E. Silicone Products Business Division
Michael L. Cuenco	Kaiser Permanente
Sharon C. Dodson	Kellogg Company
Necip Doganaksoy	General Electric Corporate Research and Development
Melissa Durfee	Wyman-Gordon Company
Luis Escobar	Louisiana State University
Leslie Fowler	Motorola Inc.
Kevin Franklin	Lockheed Aeronautical Systems Company
Paul Hamilton	Boeing
Chris Handorf	Motorola Inc.
Homer Hegedus	Motorola Inc.
Bill Henley	Chrysler Huntsville Electronics Division
Jason C. Hsu	The Ohio State University
Norio Irikura	Nippondenso Co., Ltd.
Bill Kahn	W. L. Gore & Associates
Doug Matlock	Motorola Inc.
William Meeker	Iowa State University
John Mikolaj	Union Carbide
Peter R. Nelson	Clemson University
Wayne Nelson	Consultant
Yasuo Ohashi	University of Tokyo
Joe Perry	Boeing
Jose Ramirez	W. L. Gore & Associates
Rod Reish	G. E. Silicone Products Business Division
James Sattler	Syntex Research
Robert J. Scharl	LTV Steel Company
Suzanne Scott	Texas Instruments
Mark A. Soboslai	Wheeling-Pittsburgh Steel Corporation
Jan van Schaik	The Upjohn Company
John H. Sheesley	Air Products and Chemicals, Inc.
Wayne E. Stevenson	Dow Corning Corporation
Pat Sullivan	Cameron Iron Works, Inc.
Bob Teasley	Bethlehem Steel Corporation
H. C. M. van der Knaap	Unilever Research Laboratory
Lonnie C. Vance	General Motors Corporation
Teresa Vincel	Bethlehem Steel Corporation
Philip Whittall	Unilever Research Laboratory
Joe Wolkan	General Motors Corporation
Akira Yagi	Takenaka Komuten Co., Ltd.
Kiichiro Yamamura	Japan Air Lines Co., Ltd. (retired)

The final responsibility for the SAS System lies with SAS Institute alone. We hope that you will always let us know your opinions about the SAS System and its documentation. It is through your participation that SAS software is continuously improved.

What's New in SAS/QC 9 and 9.1

Overview

SAS/QC now includes the ANOM procedure for analysis of means, a graphical and statistical method for comparing a set of means to determine whether any of them are significantly different from the overall mean. There are also a number of enhancements to the existing CAPABILITY, OPTEX, RELIABILITY, and SHEWHART procedures.

The ADX Interface for Design of Experiments is a guided point-and-click solution for engineers, scientists, statisticians, and other researchers who collaboratively design, analyze, and interpret experiments to improve industrial processes and products. The ADX Interface supports a variety of designs, including two-level, mixed level, response surface, mixture, optimal, and split-plot.

The ADX Interface now enables you to

- create general factorial designs with factors having up to nine levels
- create two-level full factorial and minimum aberration fractional factorial generalized split-plot designs
- delete inactive factors and project a fractional-factorial design to a higher-resolution design **9.1**
- join the means in a box plot **9.1**
- show clear and aliased effects in the alias structure **9.1**
- display confidence intervals in the response calculator and experiment report **9.1**
- honor block structure in a blocked design during design randomization **9.1**

The ADX Interface is documented in the book *Getting Started with the SAS 9 ADX Interface for Design of Experiments*.

ANOM Procedure

The ANOM procedure produces analysis of means (ANOM) charts for identifying group means that differ significantly from the overall mean. An ANOM chart is similar to a control chart, with a process variable statistic (such as a mean) plotted versus a classification or group variable. Decision limits on the chart are used to determine which of the group means are significantly different. **9.1**

ANOM can be used as an alternative to analysis of variance (ANOVA) in the fixed effects situation. ANOM differs from ANOVA in that it identifies the groups that are different; ANOVA only determines whether a significant difference exists. ANOM has the additional advantage of a convenient graphical representation and lends itself

to quality improvement applications in which the end user has limited background in statistics.

PROC ANOM can be applied to both variables and attribute data, and to equal and unequal sample sizes.

CAPABILITY Procedure

You can now juxtapose displays, including box-and-whisker plots, dot plots, and carpet plots, with histograms as aids for visualizing the distribution of process data. The following options are new in the HISTOGRAM statement:

- The BMCBOXFILL= option specifies the fill color for a box-and-whisker plot in the bottom margin. By default, the box-and-whisker plot is not filled.
- The BMCFRAME= option specifies the color for filling the frame of a bottom margin plot. By default, this area is not filled.
- The BMCOLOR= option specifies the color of the dot plot, carpet plot, or the outline of a box-and-whisker plot in the bottom margin.
- The BMMARGIN= option specifies the height of a bottom margin plot.
- The BMPLOT= option produces a box-and-whisker plot, dot plot, or carpet plot along the bottom margin of a histogram. A box-and-whisker plot gives a summary of the data distribution that a histogram alone does not provide. A dot plot or carpet plot shows the distribution of individual observations.

The following options are new in the COMPHIST and HISTOGRAM statements:

- The FRONTREF option causes reference lines to be drawn in front of histogram bars.
- The LOWER= suboption in the KERNEL option specifies lower bounds for fitted kernel density estimates.
- The UPPER= suboption in the KERNEL option specifies upper bounds for fitted kernel density estimates.

9.1 CLASS= variables specified in the COMPHIST statement can now have values longer than 16 characters.

OPTEX Procedure

The OPTEX procedure now enables you to control how CLASS variables are modeled, by specifying options for the ordering of levels and the parameterization of design matrix columns associated with the levels. In addition, the default parameterization has been changed to be orthogonal, providing efficiency values and parameter variances that are easier to interpret and to compare.

The following options are new in the CLASS statement:

- The DESCENDING option reverses the sorting order of the classification levels.
- The ORDER= option specifies how the classification levels are sorted.
- The PARAM= option specifies the parameterization method for the design matrix columns associated with classification levels.
- The REF= option specifies the reference level for effect or reference parameterization.

RELIABILITY Procedure

- The MCFPLOT statement has been enhanced to allow recurrence and censoring ages to be grouped into intervals.
- The INTERPOLATE= option, in the MCFPLOT statement, enables plotted points to be connected with either a step function or a straight line.
- The PPOS=NELSONAALEN option, in the PROBPLOT statement, enables the use of Nelson-Aalen plotting positions.
- You can create probability plots when all failure modes act and plot CDF estimates for individual modes on the same plot. **9.1**
- You can compute cumulative distribution function estimates and confidence limits when all modes act. **9.1**

SHEWHART Procedure

In previous releases of the SHEWHART procedure, the data associated with a subgroup in an input summary data set had to be complete for that subgroup to be plotted. For example, if the variable _LCLX_ in a TABLE= data set contained a missing value for a given subgroup, that subgroup would not be plotted. The SHEWHART procedure now processes data and control limit values independently, so that the missing value for _LCLX_ produces a gap in the lower control limit line for that subgroup. The data, central line, and upper control limit for the subgroup are all plotted, assuming the values of the associated variables are not missing.

Phase and block variable values can now be up to 48 characters long. **9.1**

The following options are new or enhanced:

- The BOX= option in the PROC SHEWHART statement specifies an input data set containing subgroup summary data for producing schematic box charts.
- The CGRID= option specifies the color of horizontal grid lines positioned at labeled major tick marks.
- The CLABEL= option specifies the text color for labels produced by the ALLLABEL=, ALLLABEL2=, OUTLABEL=, and OUTLABEL2= options.
- The CTESTLABBOX= option specifies the text color for non-overlapping labels for positive tests for special causes.

- The CTESTSYM= option specifies the color of the symbol used to plot subgroups with positive tests for special causes.
- The HTML2= option enables you to associate URLs with subgroup points on a secondary chart when graphics output is directed to HTML.
- The LABELANGLE= option specifies the angle at which labels produced by the ALLLABEL=, ALLLABEL2=, OUTLABEL=, and OUTLABEL2= options are drawn.
- The OUTBOX= option in the BOXCHART statement produces an output data set containing complete subgroup summary data for schematic box charts.

9.1

- The PHASEVARLABEL option displays the phase variable label above the phase values.

9.1

- The PHASEVALSEP option draws a vertical line separating phase values.
- The SMETHOD=MVGRANGE option in the XRCHART statement causes the process standard deviation to be estimated using a moving range of subgroup averages. You can use this to construct control charts for means when the jth measurement in the ith subgroup can be modeled as $x_{ij} = \sigma_B \omega_i + \sigma_W \epsilon_{ij}$, where σ_B^2 is the between-subgroup variance, σ_W^2 is the within-subgroup variance, the ω_i are independent with zero mean and unit variance, and the ω_i are independent of the ϵ_{ij}. This method can also be used to construct the three-way control chart, which is advocated for this situation by Wheeler (1995). A three-way control chart is useful when sampling, or within-group, variation is not the only source of variation, as discussed in "Multiple Components of Variation" on page 2009. A three-way control chart comprises a chart of subgroup means, a moving range chart of the subgroup means, and a chart of subgroup ranges. When you specify the SMETHOD=MVGRANGE option, the XRCHART statement produces the appropriate charts of subgroup means and subgroup ranges.
- The TESTLABBOX option produces labels for subgroups with positive tests for special causes that are positioned so they do not overlap, if possible. The labels are enclosed in boxes which are connected to the associated subgroup points with line segments.
- The TESTSYM= option specifies a symbol for plotting subgroups with positive tests for special causes.
- The TESTSYMHT= option specifies the height of the symbol used to plot subgroups with positive tests for special causes.
- The WNEEDLES= option specifies the width, in pixels, of needles that connect plotted points to the central line.

9.1

- The ZEROSTD=NOLIMITS option suppresses the degenerate control limits on a control chart produced when the estimated process standard deviation $\hat{\sigma}$ is zero.

References

Wheeler, D. J. (1995), *Advanced Topics in Statistical Process Control*, Knoxville, TN: SPC Press, Inc.

Using This Book

Overview

The *SAS/QC User's Guide* provides complete documentation, including introductory examples, syntax, computational details, and advanced examples for the procedures in SAS/QC 9.1. In general, this book can be used for all current releases of SAS/QC software, and it replaces and updates the information provided by *SAS/QC Software: User's Guide, Version 8, Volumes 1, 2, and 3* and *SAS/QC Software: Changes and Enhancements for Release 8.1*.

Point-and-click interfaces for basic statistical quality improvement methods and design of experiments are also included in SAS/QC software. The SQC Menu System for statistical quality control applications is described in *SAS/QC Software: SQC Menu System, Version 6, First Edition*. The ADX Interface for the design and analysis of experiments is described in *Getting Started with the SAS ADX Interface for Design of Experiments.*

Organization

This book is divided into parts, each of which corresponds to a procedure in SAS/QC software and contains one or more chapters. For example, the part describing the CAPABILITY procedure contains a chapter for each plot statement (such as the HISTOGRAM statement) in the procedure. Similarly, the part describing the SHEWHART procedure contains a chapter for each chart statement (such as the XRCHART statement) in the procedure. The following list summarizes the types of information provided for each procedure:

Overview provides a general description of what the procedure does.

Getting Started illustrates simple uses of the procedure using tutorial examples.

Syntax constitutes the major reference section for the syntax of the procedure. First, the statement syntax is summarized. Next, functional summary tables list the options classified by function. Finally, a dictionary of options, listed in alphabetical order, provides details on each option.

Details describes features of the procedure, including equations, computational methods, and input and output data sets.

Examples provides examples that illustrate common and advanced applications of the procedure.

References lists books and journal articles relevant to the procedure.

Typographical Conventions

SAS/QC User's Guide uses various type styles, as explained in the following list:

roman	is the standard type style used for most text.
UPPERCASE ROMAN	is used for SAS statements, variable names, and SAS language elements when they appear in the text. However, you can enter these elements in your own SAS code in lowercase, uppercase, or a mixture of the two. This style is also used for identifying arguments and values (in the Syntax specifications) that are literals (for example, to denote valid keywords for a specific option).
UPPERCASE BOLD	is used in the "Syntax" section to identify SAS keywords such as the names of procedures, statements, and options.
italic	is used for user-supplied values for options. It is also used for terms that are defined in the text, for emphasis, and for references.
`monospace`	is used to show examples of SAS statements. In most cases, this book uses lowercase type for SAS code. You can enter your own SAS code in lowercase, uppercase, or a mixture of the two. This style is also used for values of character variables when they appear in the text.

Conventions for Examples

Most of the output shown in this book is produced with the following SAS System options:

```
options linesize=80 pagesize=76 nonumber nodate;
```

The template STATDOC.TPL is used to create the HTML output that appears in the online (CD) version. A style template controls stylistic HTML elements such as colors, fonts, and presentation attributes. The style template is specified in the ODS HTML statement as follows:

```
ODS HTML style=statdoc;
```

If you run the examples, you may get slightly different output. This is a function of the SAS System options used and the precision used by your computer for floating-point calculations.

The following GOPTIONS statement is used to create the online (color) version of the graphic output.

```
filename GSASFILE 'file-specification';
goptions gsfname=GSASFILE   gsfmode =replace
         fileonly
         transparency       dev     = gif
         ftext    = swiss   lfactor = 1
         htext    = 4.0pct  htitle  = 4.5pct
         hsize    = 5.625in vsize   = 3.5in
         noborder           cback   = white
         horigin = 0in      vorigin = 0in ;
```

The following GOPTIONS statement is used to create the black-and-white version of the graphic output, which appears in the printed version of the manual.

```
filename GSASFILE '<file-specification>';

goptions gsfname=GSASFILE   gsfmode =replace
         gaccess = sasgaedt fileonly
         transparency       dev     = pslepsf
         ftext    = swiss   lfactor = 1
         htext    = 3.0pct  htitle  = 3.5pct
         hsize    = 5.625in vsize   = 3.5in
         border             cback   = white
         horigin = 0in      vorigin = 0in ;
```

Colors specified in example statements were remapped to a gray scale in the output.

Accessing the SAS/QC Sample Library

The SAS/QC sample library includes many examples that illustrate the use of SAS/QC software, including the examples used in this documentation. To access these sample programs, select **Help** from the pull-down menu in the SAS windowing environment and select **SAS Help and Documentation**. From the **Contents** list, choose **Learning to Use SAS** and **Sample SAS Programs**. From the **SAS Sample Library** choose **SAS/QC** to bring up a list of sample programs.

Online Documentation

You can access online documentation about SAS/QC software in two ways, depending on whether you are using the SAS windowing environment in the command line mode or the pull-down menu mode.

If you are using a command line, you can access the SAS/QC help menus by typing **help qc** on the SAS windowing environment command line. If you are using the SAS windowing environment with pull-down menus, you can select **SAS Help and Documentation** from the **Help** menu. Under the **Contents** tab select **SAS/QC** and then select **SAS/QC User's Guide** from the list of available topics.

Overview of SAS/QC Software

SAS/QC software, a component of the SAS System, provides a comprehensive set of tools for statistical quality improvement. You can use these tools to

- organize quality improvement efforts
- design industrial experiments for product and process improvement
- apply Taguchi methods for quality engineering
- establish statistical control of a process
- maintain statistical control and reduce variation
- analyze process capability
- develop and evaluate acceptance sampling plans

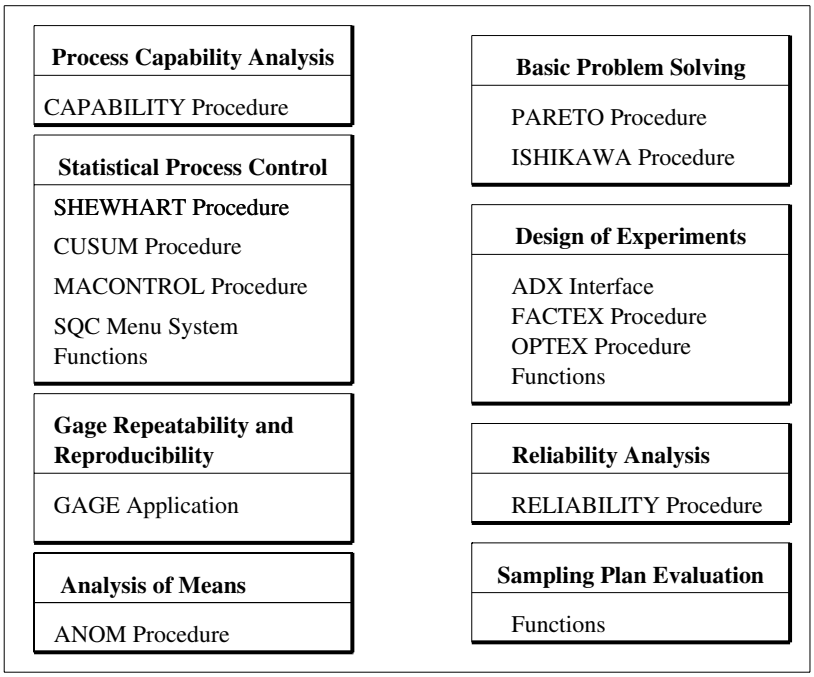
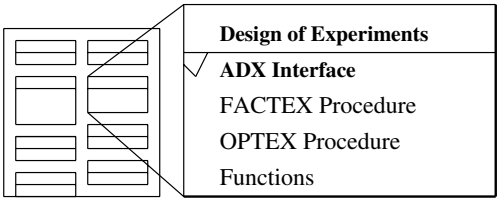


Figure 1. Components of SAS/QC Software

There are two types of tools in SAS/QC software: interfaces and procedures.

- The interfaces are complete, full-screen-oriented environments for statistical quality-improvement applications. Unlike with the procedures, using the interfaces requires no knowledge of SAS programming. These include the SQC menu system and the ADX interfaces for statistical quality-control applications.
- The procedures in SAS/QC software offer greater flexibility and power than the interface. To use a procedure, you must have a basic knowledge of the SAS language and the syntax of the procedure.

ADX Interface for Design of Experiments



The ADX Interface provides a solution for engineers and researchers who require a point-and-click interface for designing and analyzing experimental designs.

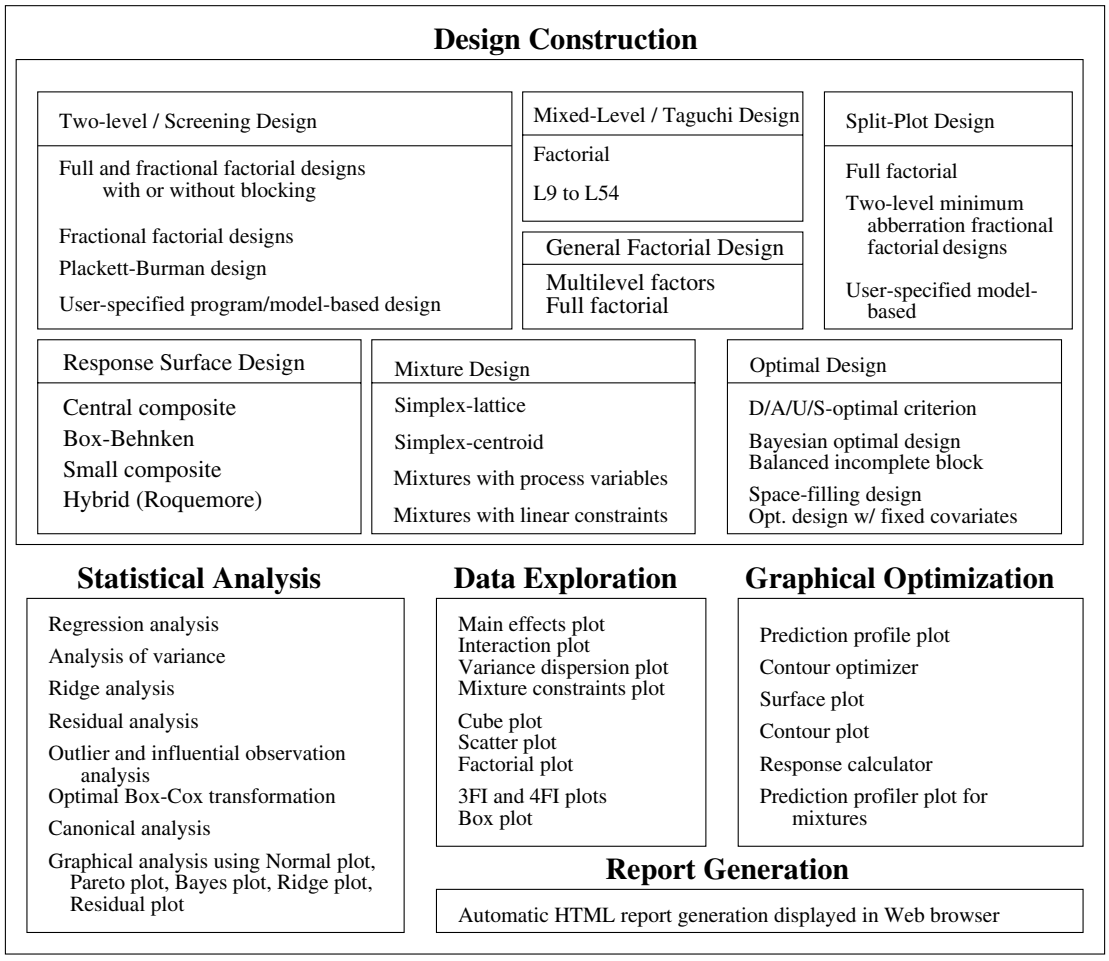
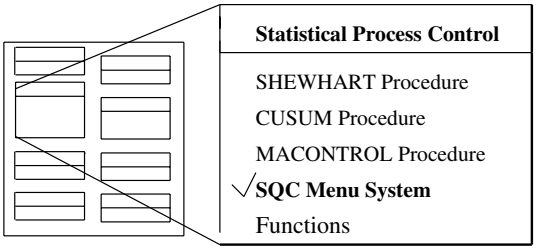


Figure 2. General Design and Analysis Facilities

Note: For more information about the ADX Interface, see *Getting Started with the SAS ADX Interface for Design of Experiments.*

SQC Menu System for Statistical Quality Control



The SQC Menu System provides facilities for standard statistical quality-control applications and is intended for quality analysts and quality-control managers, rather than for statisticians.

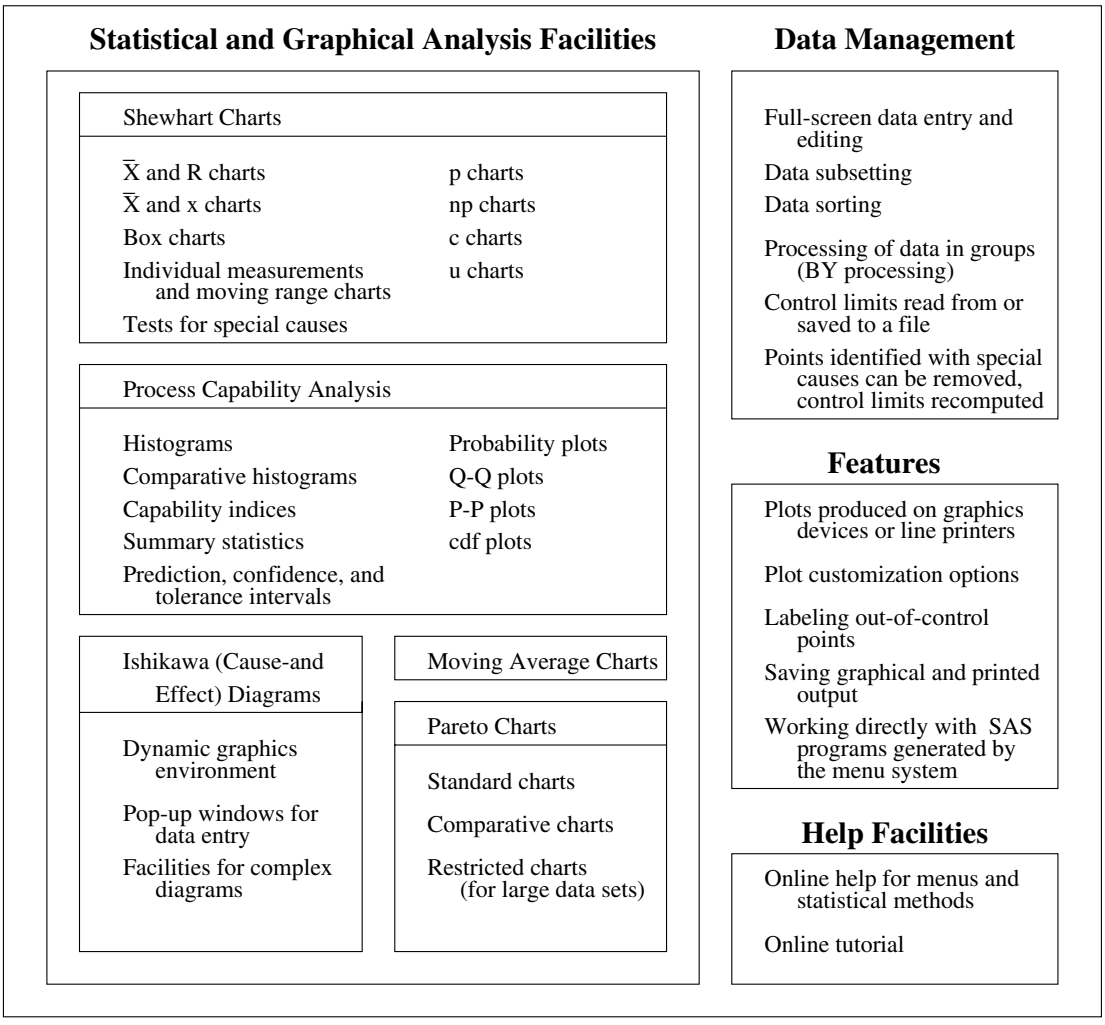
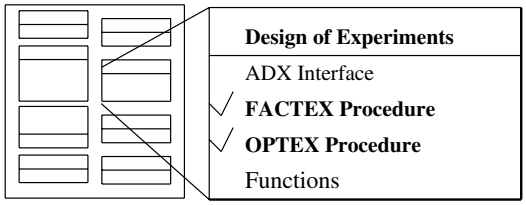


Figure 3. Overview of the SQC Menu System

Note: The SQC Menu System is documented in *SAS/QC Software: SQC Menu System for Quality Improvement, Version 6, Second Edition*.

Procedures for Design of Experiments



The FACTEX procedure constructs factorial experimental designs, which are useful for studying the effects of various factors on a response. The OPTEX procedure searches for optimal designs in situations in which standard designs are not available.

FACTEX Procedure

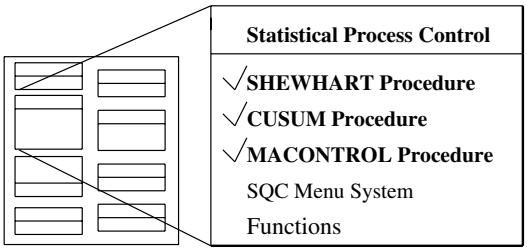
Overview	Features
Constructs orthogonally confounded full and fractional factorial designs, with and without blocking, including 2^k orthogonal array designs Constructs q^k factorial designs, where q is prime. These can be used with the SAS DATA step to construct many complex desgins, such as mixed-level factorial and balanced incomplete block designs.	No limitation on the number of factors or factor levels Approximate confounding rules are given for model constraints Designs can be chosen to have minimum size, maximum resolution, maximum block size, minimum aberration

OPTEX Procedure

Overview	Features
Searches for D-optimal and A-optimal designs Intended for situations in which a standard design is not appropriate. For example, Not all combinations of factor levels are feasible Resources restrict the number of runs The experimentation region is irregular The model is nonlinear or nonstandard linear	Add points to an existing design Augment preexisting covariates Evaluate an existing design Improve an initial design Various search algorithms and selection criteria are available Flexible GLM-type models are available

Figure 4. Overview of the Experimental Design Procedures

Procedures for Control Chart Analysis



The SHEWHART procedure creates all commonly encountered Shewhart charts for variables and attributes. The CUSUM procedure creates cumulative sum control charts. The MACONTROL procedure creates moving average charts.

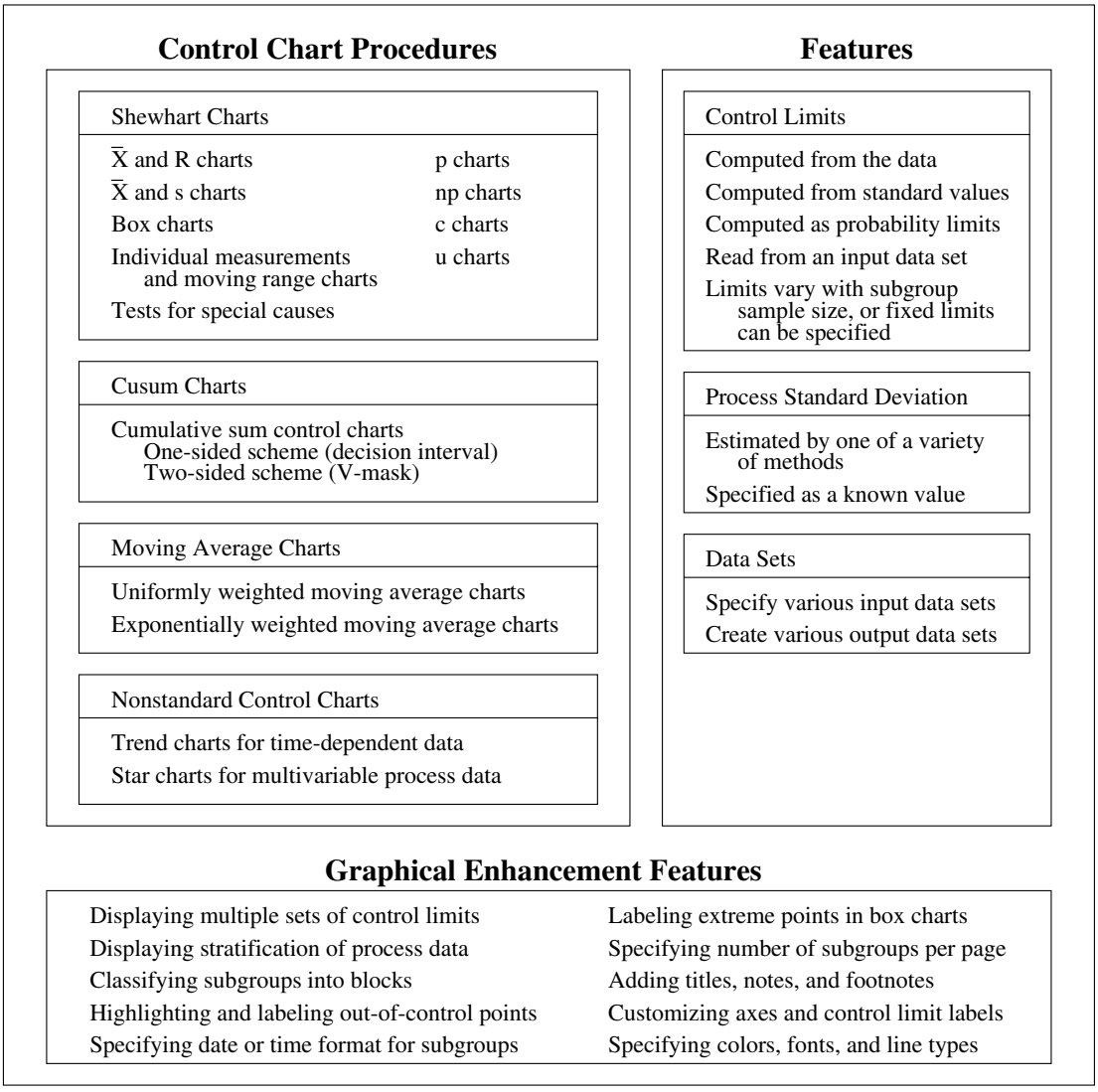
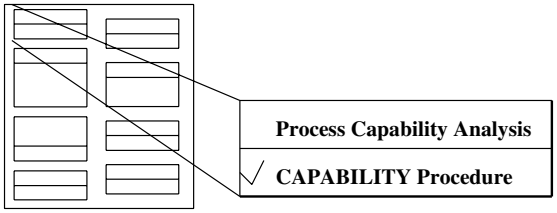


Figure 5. Overview of Control Chart Analysis Procedures

Procedure for Process Capability Analysis



The CAPABILITY procedure compares the distribution of output from an in-control process to the specification limits of the process to determine the consistency with which the specification limits can be met.

CAPABILITY Procedure

Overview

Provides graphical and statistical tools for process capability analysis

Statistical Methods

- Descriptive Statistics
- Capability Indices: Cp, Cpk, CPL, CPU, Cpm, k
- Confidence, tolerance, and prediction intervals
- Goodness-of-fit based on ecdf
- Estimation of areas outside specification limits

Other Graphical Features

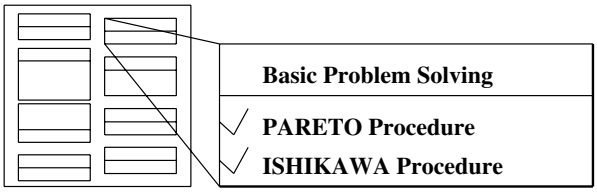
- Inset boxes with summary statistics
- Customized axes and legends
- Titles, footnotes, notes
- Colors, fonts, symbol markers

Graphical Methods

- Histograms, superimposed with
  - specification limits
  - fitted pdf curves
  - kernel density estimates
  - summary statistics
- Comparative histograms, superimposed with
  - specification limits
  - fitted pdf curves
  - kernel density estimates
  - summary statistics
- cdf plots, superimposed with
  - specification limits
  - fitted theoretical distribution curves
- Probability plots, superimposed with
  - specification limits
- Quantile-quantile (Q-Q) plots, superimposed with specification limits
- Probability-probability (P-P) plots
- Available distributions
  - normal
  - lognormal
  - exponential
  - Weibull
  - gamma
  - beta

Figure 6. Overview of Process Capability Analysis Procedure

Procedures for Basic Quality Problem Solving



The PARETO procedure creates charts that display the relative frequency of problems in a process or operation. The ISHIKAWA procedure creates a cause-and-effect or fishbone diagram, which displays factors that affect a quality characteristic or problem.

PARETO Procedure

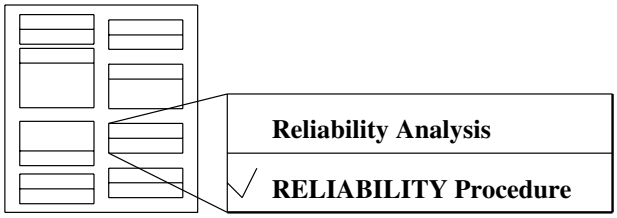
Overview	Graphical Features
Creates basic and comparative Pareto Charts	Analysis based on counts, percentages, or weights Cumulative percent curve Comparisons across classification variables Color and pattern highlighting options for bars Labels for bars and sample sizes Customized axes and legends Titles, footnotes, notes Colors, fonts, symbol markers
Other Features	
Input raw or categorized data Line printers or graphics devices Restricted Pareto charts (for large data sets)	

ISHIKAWA Procedure

Overview	Features for Complex Diagrams
Provides a mouse-driven environment for creating Ishikawa diagrams (cause-and-effect diagrams)	Zooming sections in separate windows Merging subdiagrams Dynamic foliation and defoliation Highlighting critical elements Pop-up notepads for diagram elements
Interactive Graphics Environment	**Options for Saving Diagrams**
Diagram elements (branches, stems) added and edited with mouse Diagram elements dynamically linked to preserve tree structure	In a SAS data set for future editing In a graphics catalog

Figure 7. Overview of Quality Problem-Solving Procedures

Procedure for Reliability Analysis



The RELIABILITY procedure provides tools for reliability and survival data analysis and for recurrence data analysis.

RELIABILITY Procedure

Overview

Provides graphical and statistical tools for the analysis of reliability data

Statistical Methods

Fitting of lifetime distributions:
- Weibull
- Lognormal
- Loglogistic
- Extreme value
- Normal
- Logistic
- 3 parameter gamma

Right, left, interval censored lifetimes

Parameter estimation by maximum likelihood and least squares

Confidence intervals for distribution parameters, percentiles, survival function

Analysis of Poisson and binomial data

Regression models

Accelerated life test models

Graphical Features

Probability plots for lifetime distributions:
- Kaplan-Meier, median ranks, expected ranks methods for right censored data
- Turnball algorithm for arbitrarily censored data

Available distributions for probability plots:
- Weibull
- Exponential
- Lognormal
- Loglogistic
- Extreme value
- Normal
- Logistic

Lifetime-stress relationship plots for accelerated lifetest data

Plots of mean cumulative function and confidence intervals for repairable systems data

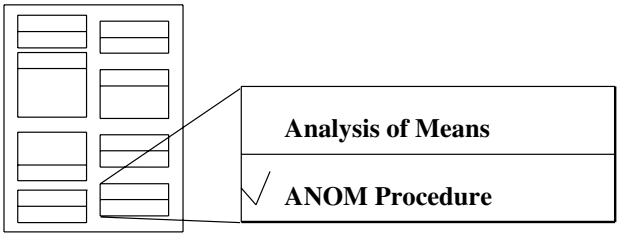
Plots for multiple populations

Other Graphical Features

Inset box with summary statistics
Customized axes and legends
Title, footnotes, notes
Colors, fonts, symbol markers

Figure 8. Overview of Reliability Analysis Procedure

Procedure for Analysis of Means



The ANOM procedure provides tools for simultaneously comparing a group of k treatment means with their overall mean at a specified significance level α. The procedure creates ANOM charts for various types of response data, including continuous measurements, proportions, and rates.

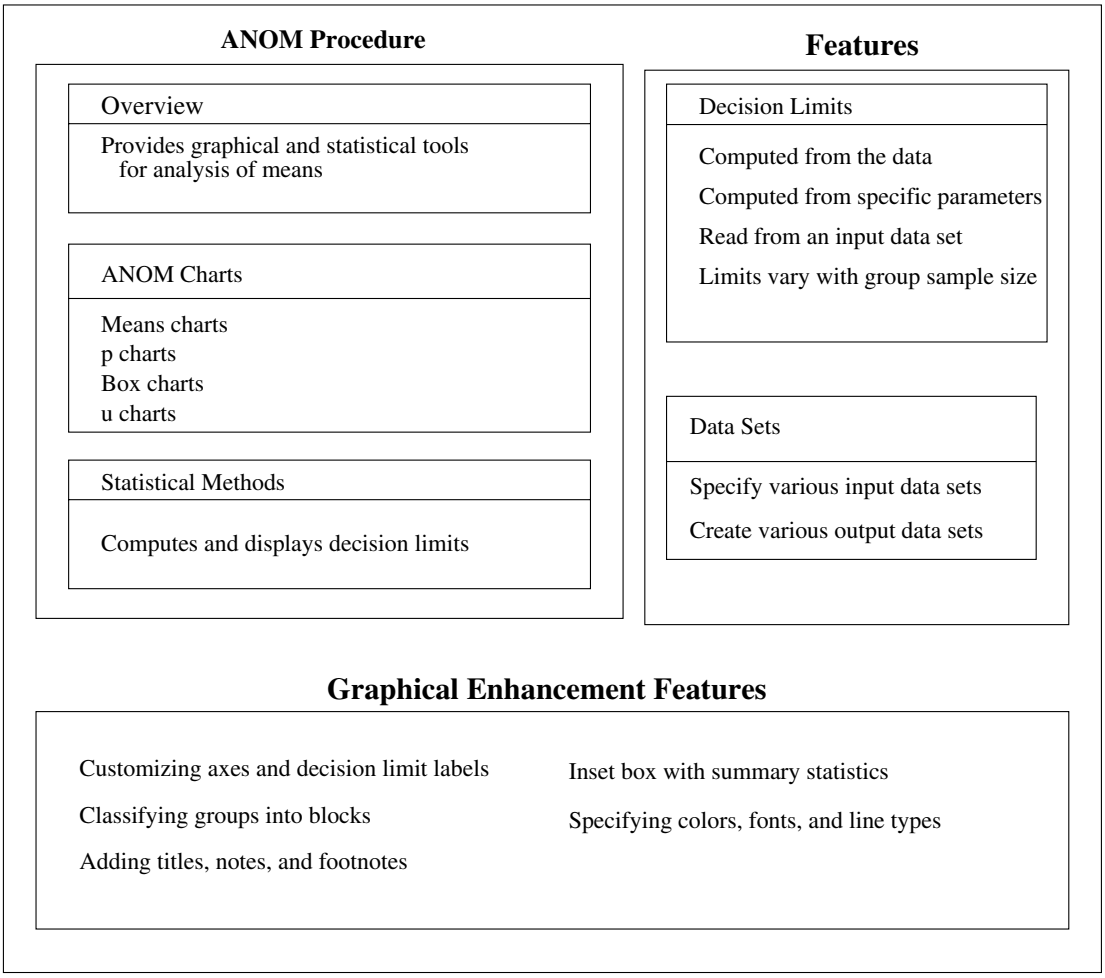


Figure 9. Overview of Analysis of Means Procedure

Part 1
The ANOM Procedure

Contents

Chapter 1. PROC ANOM and General Statements 3

Chapter 2. XCHART Statement 13

Chapter 3. PCHART Statement 53

Chapter 4. UCHART Statement 83

Chapter 5. BOXCHART Statement 109

Chapter 6. INSET Statement 141

Chapter 7. References 157

Chapter 1
PROC ANOM and General Statements

Chapter Contents

OVERVIEW 5
- Uses of Analysis of Means 5
- Terminology 6
- History 7
- Using the ANOM Procedure 7

SYNTAX OVERVIEW FOR THE ANOM PROCEDURE 8
- BY and ID Statements 8
- Graphical Enhancement Statements 9

SYNTAX FOR THE PROC ANOM STATEMENT 9

Chapter 1
PROC ANOM and General Statements

Overview

Analysis of means (ANOM) is a graphical and statistical method for simultaneously comparing k treatment means with their overall mean at a specified significance level α. You can use the ANOM procedure to create ANOM charts for various types of response data, including continuous measurements, proportions, and rates.

In addition, you can use the ANOM procedure to

- create charts from either response values or summarized data
- analyze multiple response variables
- specify decision limits in terms of the significance level (α)
- compute decision limits from the data and automatically adjust decision limits for unequal sample sizes
- save chart statistics and decision limits in output data sets
- tabulate chart statistics and decision limits.

Uses of Analysis of Means

Many statistical quality improvement applications involve a comparison of treatment means to determine which are significantly different from the overall average. For example, a manufacturing engineer might run an experiment to investigate which of six positions on a machine are producing different output, in the sense that the average measurement for each position differs from the overall average. Likewise, a health care system administrator might ask which clinics in the system have a higher or lower rate of admissions than the average for all clinics.

Questions of this type can be answered with *analysis of means*, which is an alternative to one-way analysis of variance (ANOVA) for a fixed effects model. However, unlike ANOVA, which simply determines whether there is a statistically significant difference in the treatment means, ANOM identifies the means that are significantly different. As a statistical technique, ANOM is a method for making multiple comparisons that is sometimes referred to as a "multiple comparison with the weighted mean". Analysis of means lends itself to quality improvement applications because it has a simple graphical representation that is similar to a Shewhart chart and requires little training to interpret. This representation is also useful for assessing practical significance.

Figure 1.1 illustrates a typical ANOM chart. The central line represents the overall average. The treatment means, plotted as deviations from the overall average are

compared with upper and lower decision limits to identify which are significantly different from the overall mean (in this case, the means corresponding to the first, fourth, and sixth positions).

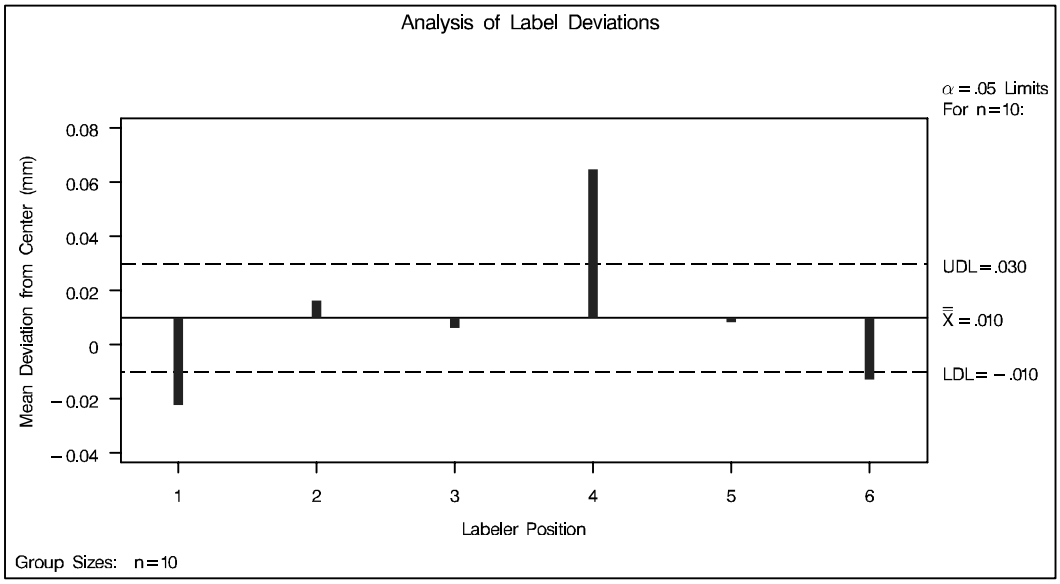


Figure 1.1. Typical ANOM Chart

Although the term "analysis of means" suggests that the method is intended for means of continuous response measurements, the method is also applicable to means of attributes data, including proportions and rates.

Analysis of means was introduced as a tool for statistical quality control by Ellis Ott in 1967, and it became popular during the early 1980s, when it was applied to experimental data in manufacturing. In this setting, measurements are taken at a number of treatment levels (factor levels). During the 1990s, the use of ANOM spread to service industry applications and, in particular, health care quality improvement. In these settings, data (such as utilization rates) are observed for a number of groups (such as hospitals or clinics).

Terminology

In order to accommodate the growing variety of modern applications for analysis of means, the term *group* is used instead of treatment level throughout the documentation for the ANOM procedure. Likewise, the term *group-variable* is used to refer to the variable in the input data set that classifies the observations into treatment levels. In the ANOM procedure, a *group-variable* plays the same role as a CLASS variable in the GLM and ANOVA procedures, and it is syntactically the same as a *subgroup-variable* in the SHEWHART procedure.

The nomenclature for ANOM charts is the same as that for Shewhart charts: $\bar{X}$ charts for means, p charts for proportions, and u charts for rates. Consequently, the syntax for the ANOM procedure is patterned after the syntax for the SHEWHART proce-

dure. However, there are some important differences between ANOM charts and Shewhart charts:

- Analysis of means is formally a test of hypothesis, whereas a Shewhart chart is used to distinguish between special and common causes of variation.
- In an ANOM chart, the horizontal axis corresponds to the *group-variable*, and it identifies the groups, which can be displayed in any order. In a Shewhart chart, the horizontal axis corresponds to the *subgroup-variable*, and it identifies the order in which the subgroup measurements were taken.
- An ANOM chart displays response summary statistics for a set of groups (treatments) at a specific time. A Shewhart chart displays subgroup summary statistics for a specific process where the subgroups are made up of measurements taken over successive points in time.
- In an ANOM chart, the decision limits are determined by a specified significance level (α), which is the probability that under the null hypothesis of no treatment differences, at least one of the response summary statistics will exceed the decision limits. In a Shewhart chart, control limits are typically computed as 3σ limits.

History

Analysis of means compares the absolute deviations of group means from their overall mean, an approach that was initially studied by Laplace in 1827. Halperin and others derived a version of this method in the form of a multiple significance test in 1955. Ott developed a graphical representation for the test and introduced the term "analysis of means" in 1967. Refer to Ott (1967) and Ott (1975).

P. R. Nelson (1982a) and L. S. Nelson (1983) provided exact critical values for ANOM when the groups have equal sample sizes. P. R. Nelson (1991) developed a method for computing exact critical values for ANOM when the group sample sizes are not equal. Refer to Nelson, Coffin, and Copeland (2003) for more information on the use of ANOM in engineering experimentation.

Using the ANOM Procedure

The PROC ANOM statement invokes the ANOM procedure and it optionally identifies various data sets.

To create an ANOM chart, you specify a chart statement (after the PROC ANOM statement) that specifies the type of ANOM chart you want to create and the variables in the input data set that you want to analyze. For example, the following statements request a basic ANOM chart for treatment means:

```
proc anom data=values;
   xchart weight*treatment;
run;
```

Here, the DATA= option specifies an input data set (values) that contains the *response* measurement variable (weight) and the *group-variable* (treatment). You can use options in the PROC ANOM statement to

- specify input data sets containing variables to be analyzed, decision limits, and annotation information
- specify a graphics catalog for saving graphical output

Note: If you are learning to use the ANOM procedure, you should read both this chapter and the "Getting Started" section in the chapter for the chart statement that corresponds to the chart you want to create.

Syntax Overview for the ANOM Procedure

The following are the primary statements that control the ANOM procedure:

```
PROC ANOM < options > ;
   XCHART (responses)/*group-variable
        < (block-variables) > < =symbol-variable > < / options >;
   PCHART (responses)/*group-variable
        < (block-variables) > < =symbol-variable > < / options >;
   UCHART (responses)/*group-variable
        < (block-variables) > < =symbol-variable > < / options >;
   BOXCHART (responses)/*group-variable
        < (block-variables) > < =symbol-variable > < / options >;
   INSET keyword-list < / options >;
```

The PROC ANOM statement invokes the procedure and specifies the input data set. The chart statements create different types of charts. You can specify one or more of each of the chart statements. For details, read the chapter on the chart statement that corresponds to the type of chart that you want to produce.

BY and ID Statements

In addition, you can optionally specify one of each of the following statements:

BY *variables* **;**

ID *variables* **;**

The BY statement specifies variables in the input data set that are used for BY processing. A separate chart is created for each set of observations defined by the levels of the BY variables. The input data set must be sorted in order of the BY variables.

The ID statement specifies variables used to identify observations. The ID variables must be variables in the DATA= or SUMMARY= input data sets.

The ID variables are used in the following ways:

- If you create an OUTSUMMARY= or OUTTABLE= data set, the ID variables are included. If the input data set is a DATA= data set, only the values of the ID variables from the first observation in each group are passed to the output data set.
- If you specify the TABLEID or TABLEALL options in a chart statement, the table produced is augmented by a column for each of the ID variables. Only the values of the ID variables from the first observation in each group are tabulated.
- If you specify the BOXSTYLE= SCHEMATICID option or the BOXSTYLE= SCHEMATICIDFAR option in the BOXCHART statement, the value of the first variable listed in the ID statement is used to label each extreme observation.

Graphical Enhancement Statements

You can use TITLE, FOOTNOTE, and NOTE statements to enhance graphical and printed output. You can also use AXIS, LEGEND, and SYMBOL statements to enhance your charts. For details, refer to *SAS/GRAPH Software: Reference* and see the chapter for the chart statement that you are using.

Syntax for the PROC ANOM Statement

The syntax for the PROC ANOM statement is as follows:

PROC ANOM *options* **;**

The PROC ANOM statement starts the ANOM procedure and optionally identifies various data sets. The following options can appear in the PROC ANOM statement.

ANNOTATE=*SAS-data-set*
ANNO=*SAS-data-set*
specifies an input data set containing ANNOTATE= variables as described in *SAS/GRAPH Software: Reference*. You can use this data set to add features to ANOM charts. Features provided in this data set are displayed on every chart produced in the current run of the ANOM procedure.

BOX=*SAS-data-set*
names an input data set that contains group summary statistics, decision limits, and outlier values in "strung out" form, with more than one observation per group. Each observation corresponds to one feature of one group's box-and-whisker plot. Typically, this data set is created as an OUTBOX= data set in a previous run of the ANOM procedure with a BOXCHART statement. The BOX= data set is the only kind of summary data set you can use to produce schematic box-and-whisker plots. The BOXCHART statement is the only chart statement you can use with a BOX= input data set.

DATA=*SAS-data-set*
names an input data set that contains response values (typically, measurements or counts) as observations. Note that the DATA= data set may need sorting. If the values of the *group-variable* are numeric, you must sort the data set so that these

values are in increasing order (within BY groups). Use PROC SORT if the data are not already sorted.

The DATA= data set may contain more than one observation for each value of the *group-variable*. This happens, for example, when you produce a chart for means and ranges with the XCHART statement.

You cannot use a DATA= data set together with a SUMMARY= or a TABLE= data set. If you do not specify one of these three input data sets, the ANOM procedure uses the most recently created data set as a DATA= data set. For more information, see the "DATA= Data Set" section in the chapter for the chart statement you are using.

GOUT=*graphics-catalog*
specifies the graphics catalog for graphics output from the ANOM procedure. This is useful if you want to save the output.

SUMMARY=*SAS-data-set*
names an input data set that contains group summary statistics. For example, you can read sample sizes, means, and standard deviations for the groups to create an ANOM chart. Typically, this data set is created as an OUTSUMMARY= data set in a previous run of the ANOM procedure, but it can also be created using a SAS summarization procedure such as PROC MEANS.

Note that the SUMMARY= data sets may need sorting. If the values of the *group-variable* are numeric, you need to sort the data set so that these values are in increasing order (within BY groups). Use PROC SORT if the data are not already sorted. The SUMMARY= data set can contain only one observation for each value for the *group-variable*.

You cannot use a SUMMARY= data set with a DATA= or a TABLE= data set. If you do not specify one of these three input data sets, the ANOM procedure uses the most recently created data set as a DATA= data set. For more information, see the "SUMMARY= Data Set" section in the chapter for the chart statement you are using.

LIMITS=*SAS-data-set*
names an input data set that contains preestablished decision limits or the parameters from which decision limits can be computed. Each observation in a LIMITS= data set provides decision limit information for a *response*. Typically, this data set is created as an OUTLIMITS= data set in a previous run of the ANOM procedure.

If you omit the LIMITS= option, then decision limits are computed from the data in the DATA= or SUMMARY= input data sets. For details about the variables needed in a LIMITS= data set, see the "LIMITS= Data Set" section in the chapter for the chart statement you are using.

TABLE=*SAS-data-set*
names an input data set that contains group summary statistics and decision limits. Each observation in a TABLE= data set provides information for a particular group and *response*. Typically, this data set is created as an OUTTABLE= data set in a previous run of the ANOM procedure.

You cannot use a TABLE= data set with a DATA= or a SUMMARY= data set. If you do not specify one of these three input data sets, the ANOM procedure uses the most recently created data set as a DATA= data set. For more information, see the "TABLE= Data Set" section in the chapter for the chart statement that you are using.

Chapter 2
XCHART Statement

Chapter Contents

OVERVIEW 15

GETTING STARTED 15
- Creating ANOM Charts for Means from Response Values 15
- Creating ANOM Charts for Means from Group Summary Data 18
- Saving Summary Statistics for Groups 20
- Saving Decision Limits 21

SYNTAX 23
- Summary of Options 24

DETAILS 31
- Constructing ANOM Charts for Means 31
- Constructing ANOM Charts for Two-Way Layouts 33
- Output Data Sets 35
- ODS Tables 37
- Input Data Sets 37
- Axis Labels 41
- Missing Values 41

EXAMPLES 41
- Example 2.1. ANOM Charts with Unequal Group Sizes 41
- Example 2.2. ANOM for a Two-Way Classification 43
- Example 2.3. ANOM Charts Using LIMITS= Data Set 46
- Example 2.4. ANOM for Cell Means in Presence of Interaction 48

Chapter 2
XCHART Statement

Overview

The XCHART statement creates an ANOM chart for group (treatment level) means of response values. You can use options in the XCHART statement to

- compute decision limits from the data based on specified parameters, such as the significance level (α)
- tabulate group sample sizes, group means, decision limits, and other information
- save decision limits in an output data set
- save group sample sizes and group means in an output data set
- read decision limits and decision limit parameters from a data set
- display distinct sets of decision limits for different sets of groups
- add block legends and symbol markers to identify special groups
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

Getting Started

This section introduces the XCHART statement with simple examples that illustrate the most commonly used options. Complete syntax for the XCHART statement is presented in the "Syntax" section on page 23, and advanced examples are given in the "Examples" section on page 41.

Creating ANOM Charts for Means from Response Values

See ANMX1 in the SAS/QC Sample Library

A manufacturing engineer carries out a study to determine the source of excessive variation in the positioning of labels on shampoo bottles. * A labeling machine removes bottles from the line, attaches the labels, and returns the bottles to the line. There are six positions on the machine, and the engineer suspects that one or more of the position heads might be faulty.

A sample of 60 bottles, 10 per position, is run through the machine. For each bottle, the deviation of the label is measured in millimeters, and the machine position is

*This example is based on a case study described by Hansen (1990).

recorded. The following statements create a SAS data set named LabelDeviations, which contains the deviation measurements for the 60 bottles:

```
data LabelDeviations;
   input Position @;
   do i = 1 to 5;
      input Deviation @;
      output;
   end;
   drop i;
   datalines;
1  -0.0239  -0.0285  -0.0300  -0.0043  -0.0362
1  -0.0422  -0.0014  -0.0647   0.0094  -0.0016
2  -0.0201  -0.0273   0.0227  -0.0332   0.0366
2   0.0438   0.0556   0.0098   0.0564   0.0182
3  -0.0073   0.0285  -0.0440  -0.0221  -0.0139
3   0.0486   0.0357   0.0235   0.0134  -0.0020
4   0.0669   0.1073   0.0597   0.0609   0.0755
4   0.0362   0.0561   0.0899   0.0418   0.0530
5   0.0368   0.0036   0.0374   0.0116  -0.0074
5   0.0250  -0.0080   0.0302  -0.0015  -0.0464
6   0.0049  -0.0384  -0.0204  -0.0049  -0.0120
6   0.0071  -0.0308   0.0017  -0.0285  -0.0070
run;
```

A partial listing of LabelDeviations is shown in Figure 2.1.

The Data Set LabelDeviations

Position	Deviation
1	-0.0239
1	-0.0285
1	-0.0300
1	-0.0043
1	-0.0362
1	-0.0422
1	-0.0014
1	-0.0647
1	0.0094
1	-0.0016
2	-0.0201
2	-0.0273
2	0.0227
2	-0.0332
2	0.0366
2	0.0438
2	0.0556
2	0.0098
2	0.0564
2	0.0182

Figure 2.1. Partial Listing of the Data Set LabelDeviations

The data set LabelDeviations is said to be in "strung-out" form, since each observation contains the position and the deviation measurement for a single bottle. The first 10 observations contain the measurements for the first position, the second 10

observations contain the measurements for the second position, and so on. Because the variable Position classifies the observations into groups (treatment levels), it is referred to as the *group-variable*. The input data set must be sorted by the group variable. The variable Deviation contains the deviation measurements and is referred to as the *response variable* (or *response* for short).

The following statements create the ANOM chart shown in Figure 2.2:

```
title 'Analysis of Label Deviations';
proc anom data=LabelDeviations;
   xchart Deviation*Position / alpha = 0.05;
   label Deviation = 'Mean Deviation from Center (mm)';
   label Position  = 'Labeler Position';
run;
```

This example illustrates the basic form of the XCHART statement. After the keyword XCHART, you specify the *response* to analyze (in this case, Deviation) followed by an asterisk and the *group-variable* (Position). Options are specified after the slash (/) in the XCHART statement. A complete list of options is presented in the "Syntax" section on page 23.

The input data set is specified with the DATA= option in the PROC ANOM statement when it contains raw measurements for the *response*.

Each point on the ANOM chart represents the average (mean) of the response measurements for a particular sample.

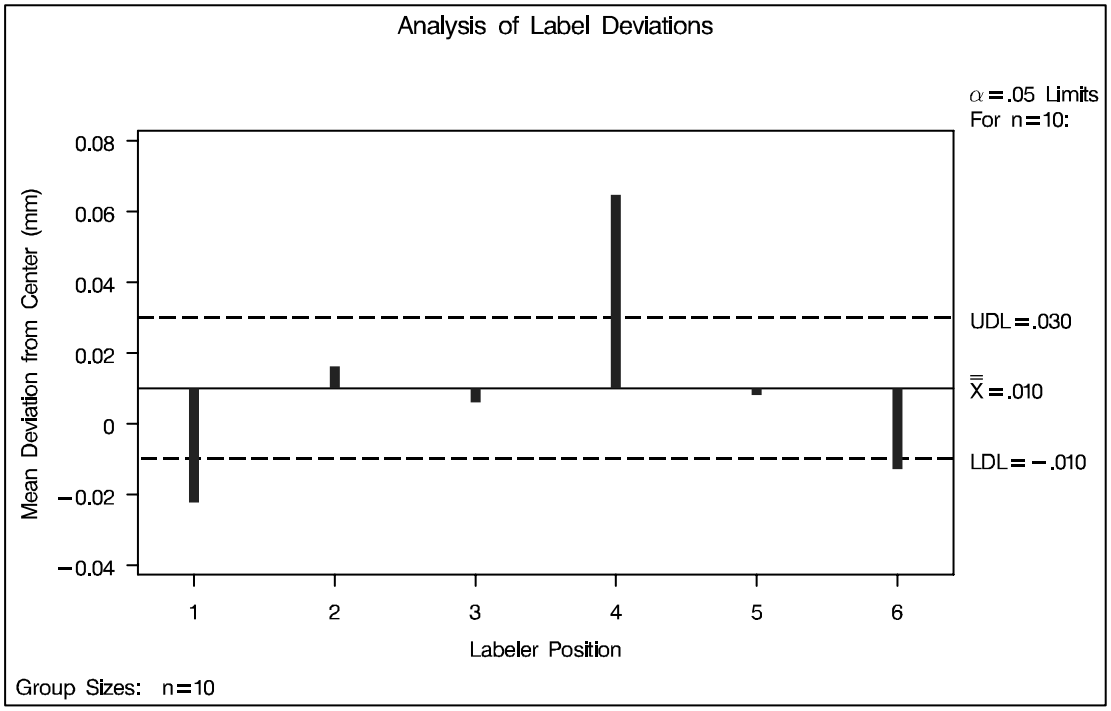


Figure 2.2. ANOM Chart for Means of Labeler Position Data

The average for Position 1 is below the lower decision limit (LDL), and the average for Position 6 is slightly below the lower decision limit. The average for Position 4 exceeds the upper decision limit (UDL). The conclusion is that Positions 1, 4, and 6 are operating differently.

By default, the decision limits shown correspond to a significance level of $\alpha = 0.05$; the formulas for the limits are given in the section "Decision Limits" on page 31. You can also read decision limits from an input data set.

For computational details, see "Constructing ANOM Charts for Means" on page 31. For details on reading raw measurements, see "DATA= Data Set" on page 37.

Creating ANOM Charts for Means from Group Summary Data

See ANMXGRP in the SAS/QC Sample Library

The previous example illustrates how you can create ANOM charts for means using measurement data. However, in many applications, the data are provided as group summary statistics. This example illustrates how you can use the XCHART statement with data of this type.

The following data set (Labels) provides the data from the preceding example in summarized form:

```
data Labels;
   input Position DeviationX DeviationS;
   DeviationN = 10;
   datalines;
1  -0.02234  0.02281
2   0.01624  0.03348
3   0.00601  0.02885
4   0.06473  0.02149
5   0.00812  0.02592
6  -0.01281  0.01597
run;
```

A listing of Labels is shown in Figure 2.3. There is exactly one observation for each group (note that the groups are still indexed by Position). The variable DeviationX contains the group means, the variable DeviationS contains the group standard deviations, and the variable DeviationN contains the group sample sizes (these are all 10).

The Data Set Labels

Position	Deviation X	Deviation S	Deviation N
1	-0.02234	0.02281	10
2	0.01624	0.03348	10
3	0.00601	0.02885	10
4	0.06473	0.02149	10
5	0.00812	0.02592	10
6	-0.01281	0.01597	10

Figure 2.3. The Summary Data Set Labels

You can read this data set by specifying it as a SUMMARY= data set in the PROC ANOM statement, as follows:

```
title 'Analysis of Label Deviations';
proc anom summary=Labels;
   xchart Deviation*Position;
run;
```

The resulting ANOM chart is shown in Figure 2.4. Note that Deviation is *not* the name of a SAS variable in the data set but is, instead, the common prefix for the names of the three SAS variables DeviationX, DeviationS, and DeviationN. The suffix characters *X*, *S*, and *N* indicate *mean*, *standard deviation*, and *sample size*, respectively. Thus, you can specify three group summary variables in a SUMMARY= data set with a single name (Deviation), which is referred to as the *response*. The name Position specified after the asterisk is the name of the *group-variable*.

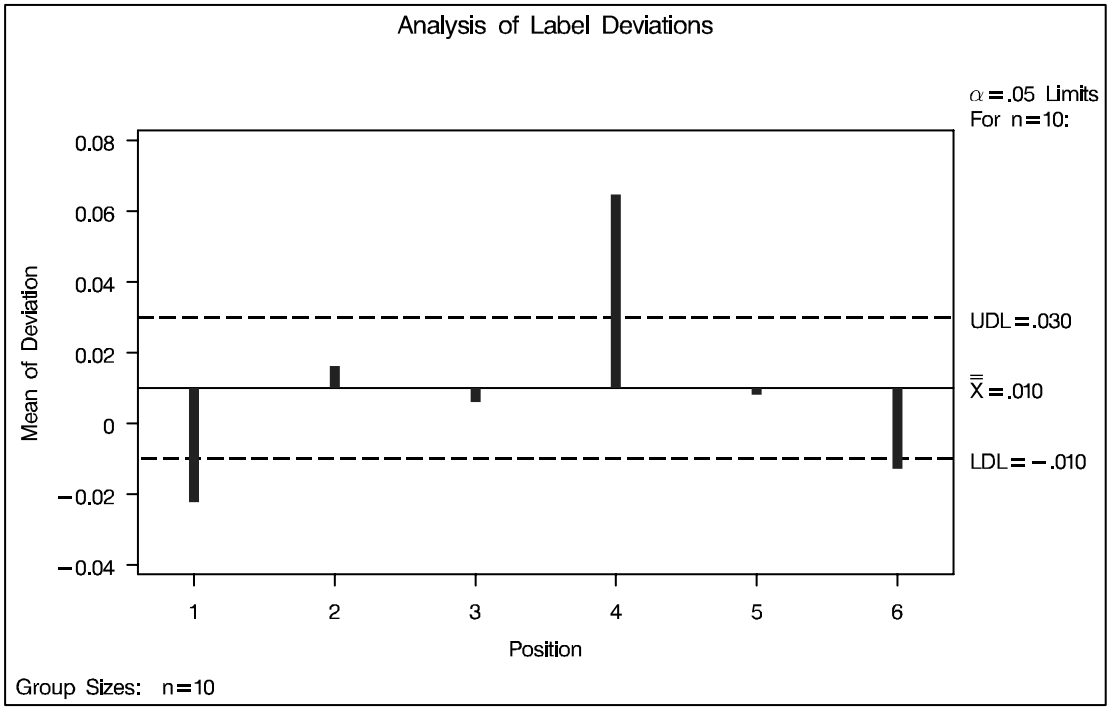


Figure 2.4. ANOM Chart for Means in Data Set Labels

In general, a SUMMARY= input data set used with the XCHART statement must contain the following variables:

- group variable
- group mean variable
- group standard deviation variable
- group sample size variable

Furthermore, the names of the group mean, standard deviation, and sample size variables must begin with the *response* name specified in the XCHART statement and end with the special suffix characters *X*, *S*, and *N*, respectively. If the names do not follow this convention, you can use the RENAME option in the PROC ANOM statement to rename the variables for the duration of the ANOM procedure step. If a label is associated with the group mean variable, it is used to label the vertical axis.

In summary, the interpretation of *response* depends on the input data set.

- If raw data are read using the DATA= option (as in the previous example), *response* is the name of the SAS variable containing the response measurements.
- If summary data are read using the SUMMARY= option (as in this example), *response* is the common prefix for the names of the variables containing the summary statistics.

For more information, see the section "SUMMARY= Data Set" on page 39.

Saving Summary Statistics for Groups

See ANMXSUM in the SAS/QC Sample Library

In this example, the XCHART statement is used to create a data set containing group summary statistics that can be read later by the ANOM procedure (as in the preceding example). The following statements read measurements from the data set LabelDeviations and create a summary data set named LabelSummary:

```
proc anom data=LabelDeviations;
   xchart Deviation*Position / outsummary=LabelSummary
                               nochart;
run;
```

The OUTSUMMARY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 2.2.

Figure 2.5 contains a listing of LabelSummary.

The Data Set LabelSummary

Position	Deviation X	Deviation S	Deviation N
1	-0.02234	0.022807	10
2	0.01625	0.033473	10
3	0.00604	0.028849	10
4	0.06473	0.021495	10
5	0.00813	0.025928	10
6	-0.01283	0.015986	10

Figure 2.5. The Summary Data Set LabelSummary

There are four variables in the data set LabelSummary.

- Position identifies the group.
- DeviationX contains the group means.
- DeviationS contains the group standard deviations.
- DeviationN contains the group sizes.

Note that the summary statistic variables are named by adding the suffix characters *X*, *S*, and *N* to the *response* Deviation specified in the XCHART statement. In other words, the variable naming convention for OUTSUMMARY= data sets is the same as that for SUMMARY= data sets.

For more information, see the section "OUTSUMMARY= Data Set" on page 36.

Saving Decision Limits

See ANMXLIM in the SAS/QC Sample Library

You can save the decision limits for an ANOM chart, together with the parameters used to compute the limits, in a SAS data set.

The following statements read measurements from the data set LabelDeviations (see the section "Creating ANOM Charts for Means from Response Values" on page 15) and save the decision limits displayed in Figure 2.2 in a data set named LabelLimits:

```
proc anom data=LabelDeviations;
   xchart Deviation*Position / outlimits=LabelLimits
                               nochart;
run;
```

The OUTLIMITS= option names the data set containing the decision limits, and the NOCHART option suppresses the display of the chart. The data set LabelLimits is listed in Figure 2.6.

Decision Limits for Labler Position Deviations

VAR	_GROUP_	_TYPE_	_LIMITN_	_ALPHA_	_LDLX_
Deviation	Position	ESTIMATE	10	0.05	-.009875608

MEAN	_UDLX_	_MSE_	_DFE_	_LIMITK_
.009996667	0.029869	.000643787	54	6

Figure 2.6. The Data Set LabelLimits Containing Decision Limit Information

The data set LabelLimits contains one observation with the limits for *response* Deviation. The values of _LDLX_ and _UDLX_ are the lower and upper decision limits for the means, and the value of _MEAN_ is the weighted average of the group means, which is represented by the central line.

The values of _MEAN_, _MSE_, _DFE_, _LIMITN_, _LIMITK_, and _ALPHA_ are the parameters used to compute the decision limits as described in the section "Constructing ANOM Charts for Means" on page 31. The value of

MSE is the mean square error, and the value of _DFE_ is the associated degrees of freedom. The value of _LIMITN_ is the nominal sample size (n) associated with the decision limits, the value of _LIMITK_ is the number of groups (k), and the value of _ALPHA_ is the value of the significance level (α). The variables _VAR_ and _GROUP_ are bookkeeping variables that save the *response* and *group-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _MSE_ are estimates computed from the data or standard (known) values specified with procedure options. In most applications, the value of _TYPE_ will be ESTIMATE.

See ANMXTAB in the SAS/QC Sample Library

You can create an output data set containing both decision limits and group summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc anom data=LabelDeviations;
   xchart Deviation*Position / outtable=LabelTab
                               nochart;
run;
```

The data set LabelTab is listed in Figure 2.7.

Summary Statistics and Decision Limits

VAR	Position	_ALPHA_	_LIMITN_	_LIMITK_	_SUBX_	_LDLX_	_MEAN_	_UDLX_	_EXLIM_
Deviation	1	0.05	10	10	-.009875608	-0.02234	.009996667	0.029869	LOWER
Deviation	2	0.05	10	10	-.009875608	0.01625	.009996667	0.029869	
Deviation	3	0.05	10	10	-.009875608	0.00604	.009996667	0.029869	
Deviation	4	0.05	10	10	-.009875608	0.06473	.009996667	0.029869	UPPER
Deviation	5	0.05	10	10	-.009875608	0.00813	.009996667	0.029869	
Deviation	6	0.05	10	10	-.009875608	-0.01283	.009996667	0.029869	LOWER

Figure 2.7. The Data Set LabelTab

This data set contains one observation for each group sample. The variables _SUBX_ and _SUBN_ contain the group means and sample sizes. The variables _LDLX_ and _UDLX_ contain the lower and upper decision limits, and the variable _MEAN_ contains the central line. The variables _VAR_ and Position contain the *response* name and values of the *group-variable*, respectively. For more information, see the section "OUTTABLE= Data Set" on page 36.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read LabelTab and display an ANOM chart (not shown here) identical to the chart in Figure 2.2:

```
title 'Analysis of Label Deviations';
proc anom table=LabelTab;
   xchart deviation*position;
label _SUBX_ = 'Mean Deviation from Center (mm)';
run;
```

Because the ANOM procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized ANOM charts.

For more information, see the section "TABLE= Data Set" on page 40.

Syntax

The basic syntax for the XCHART statement is as follows:

XCHART *response*******group-variable* **;**

The general form of this syntax is as follows:

XCHART (*responses***)****group-variable* <**(***block-variables* **)** > < =*symbol-variable* > < **/** *options* >**;**

You can use any number of XCHART statements in the ANOM procedure. The components of the XCHART statement are described as follows.

response
responses
identify one or more responses to be analyzed. The specification of *response* depends on the input data set specified in the PROC ANOM statement.

- If response values (raw data) are read from a DATA= data set, *response* must be the name of the variable containing the values. For an example, see the section "Creating ANOM Charts for Means from Response Values" on page 15.
- If summary data are read from a SUMMARY= data set, *response* must be the common prefix of the summary variables in the SUMMARY= data set. For an example, see the section "Creating ANOM Charts for Means from Group Summary Data" on page 18.
- If summary data and decision limits are read from a TABLE= data set, *response* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see the section "Saving Decision Limits" on page 21.

A *response* is required. If you specify more than one response, enclose the list in parentheses. For example, the following statements request distinct ANOM charts for the means of WEIGHT, LENGTH, and WIDTH:

```
proc anom data=measures;
   xchart (weight length width)*day;
run;
```

group-variable
is the variable that identifies groups in the data. The *group-variable* is required. In the preceding XCHART statement, DAY is the group variable.

block-variables
are optional variables that group the data into blocks of consecutive groups. The

blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker used to plot the means. Distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOLn statements.

options
enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The "Summary of Options" section, which follows, lists all options by function.

Summary of Options

The following tables list the XCHART statement options by function. Many of these options are identical to options in the SHEWHART procedure, which are described in detail beginning on page 1851 in Chapter 53, "Dictionary of Options."

Table 2.1. Tabulation Options

TABLE	creates a basic table of group means, group sample sizes, and decision limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, and TABLEOUTLIM
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLEOUTLIM	augments basic table with columns indicating decision limits exceeded

Note that specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only, that is, groups for which the mean exceeds the decision limits.

Table 2.2. Reference Line Options

CHREF=*color*	specifies color for lines requested by HREF= option
CVREF=*color*	specifies color for lines requested by VREF= option
HREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on ANOM chart
HREFDATA= *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on ANOM chart
HREFLABELS= *'label1'...'labeln'*	specifies labels for HREF= lines
HREFLABPOS=*n*	specifies position of HREFLABELS= labels
LHREF=*linetype*	specifies line type for HREF= lines
LVREF=*linetype*	specifies line type for VREF= lines
NOBYREF	specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to vertical axis on ANOM chart
VREFLABELS= *'label1'...'labeln'*	specifies labels for VREF= lines
VREFLABPOS=*n*	position of VREFLABELS= labels

Table 2.3. Block Variable Legend Options

BLOCKLABELPOS= *keyword*	specifies position of label for *block-variable* legend
BLOCKLABTYPE= *value*|*keyword*	specifies text size of *block-variable* legend
BLOCKPOS=*n*	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=*color*	specifies color for filling background in *block-variable* legend
CBLOCKVAR=*variable*| (*variables*)	specifies one or more variables whose values are colors for filling background of *block-variable* legend

Table 2.4. Axis and Axis Label Options

CAXIS=*color*	specifies color for axis lines and tick marks
CFRAME=*color*|(*color-list*)	specifies fill colors for frame for plot area
CTEXT=*color*	specifies color for tick mark values and axis labels
HAXIS=*values*|AXIS*n*	specifies major tick mark values for horizontal axis
HEIGHT=*value*	specifies height of axis label and axis legend text
HMINOR=*n*	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=*value*	specifies length of offset at both ends of horizontal axis
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent group values is to be labeled on horizontal axis
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=*n*	specifies thinning factor for tick mark labels on horizontal axis
SPLIT='*character*'	specifies splitting character for axis labels
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=*values*|AXIS*n*	specifies major tick mark values for vertical axis of ANOM chart
VMINOR=*n*	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=*value*	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis for ANOM chart
WAXIS=*n*	specifies width of axis lines

Table 2.5. Options for Specifying Parameters for Decision Limits

ALPHA=*value*	specifies the probability of a Type I error
DFE=*number*	specifies the degrees of freedom associated with the root mean square error
LIMITK=*k*	specifies number of groups for decision limits
LIMITN=*n*|VARYING	specifies either a nominal sample size for fixed decision limits or varying limits
MEAN=*value*	specifies the mean
NOREADLIMITS	computes decision limits for each *response* from the data rather than a LIMITS= data set
READINDEXES=ALL|'*label1*'...'*labeln*'	reads multiple sets of decision limits for each *response* from a LIMITS= data set
MSE=*value*	specifies the mean square error
TYPE=*keyword*	identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set

Table 2.6. Options for Displaying Decision Limits

CINFILL=*color*	specifies color for area inside decision limits
CLIMITS=*color*	specifies color of decision limits, central line, and related labels
LDLLABEL='*label*'	specifies label for lower decision limit
LIMLABSUBCHAR='*character*'	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the decision limit
LLIMITS=*linetype*	specifies line type for decision limits
NDECIMAL=*n*	specifies number of digits to right of decimal place in default labels for decision limits and central line
NOCTL	suppresses display of central line
NOLDL	suppresses display of lower decision limit
NOLIMITLABEL	suppresses labels for decision limits and central line
NOLIMITS	suppresses display of decision limits
NOLIMITSFRAME	suppresses default frame around decision limit information when multiple sets of decision limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for decision limits
NOUDL	suppresses display of upper decision limit
UDLLABEL='*string*'	specifies label for upper decision limit
WLIMITS=*n*	specifies width for decision limits and central line
XSYMBOL='*string*'|*keyword*	specifies label for central line

Table 2.7. Options for Plotting and Labeling Points

ALLLABEL=VALUE|(*variable*)	labels every point on ANOM chart
CCONNECT=*color*	specifies color for line segments connecting points on chart
CFRAMELAB=*color*	specifies fill color for frame around labeled points
CNEEDLES=*color*	specifies color for needles that connect points to central line
COUT=*color*	specifies color for portions of line segments that connect points outside decision limits
COUTFILL=*color*	specifies color for shading areas between connected points and decision limits outside the limits
NONEEDLES	suppresses vertical needles connecting points to central line
OUTLABEL=VALUE|(*variable*)	labels points outside decision limits
SYMBOLLEGEND=NONE|*name*	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=*keyword*	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=*n*	specifies width of needles

Table 2.8. Clipping Options

CCLIP=*color*	specifies color for plot symbol for clipped points
CLIPFACTOR=*value*	determines extent to which extreme points are clipped
CLIPLEGEND='*string*'	specifies text for clipping legend
CLIPLEGPOS=*keyword*	specifies position of clipping legend
CLIPSUBCHAR='*character*'	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=*symbol*	specifies plot symbol for clipped points
CLIPSYMBOLHT=*value*	specifies symbol marker height for clipped points

Table 2.9. Phase Options

CPHASELEG=*color*	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE='*string*'	specifies value of _PHASE_ in the OUTSUMMARY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=*value*|*keyword*	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels decision limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES= ALL|'*label1*'...'*labeln*'	specifies *phases* to be read from an input data set

Table 2.10. Options for Interactive ANOM Charts

HTML=(*variable*)	specifies a variable whose values are URLs to be associated with groups
HTML_LEGEND=(*variable*)	specifies a variable whose values are URLs to be associated with symbols in the symbol legend
WEBOUT=*SAS-data-set*	creates an OUTTABLE= data set with additional graphics coordinate data

Table 2.11. Output Data Set Options

OUTSUMMARY=*SAS-data-set*	creates output data set containing group summary statistics
OUTINDEX='*string*'	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=*SAS-data-set*	creates output data set containing decision limits
OUTTABLE=*SAS-data-set*	creates output data set containing group summary statistics and decision limits

Table 2.12. Grid Options

GRID	adds grid to ANOM chart
LGRID=*linetype*	specifies line type for grid requested with the GRID option
WGRID=*n*	specifies width of grid lines

Table 2.13. Plot Layout Options

ALLN	plots means for all groups
BILEVEL	creates ANOM charts using half-screens and half-pages
EXCHART	creates ANOM charts for a response only when a group mean exceeds the decision limits
MAXPANELS=*n*	maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed decision limits
NOCHART	suppresses creation of chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for group sample sizes
NPANELPOS=*n*	specifies number of group positions per panel on each chart
REPEAT	repeats last group position on panel as first group position of next panel
TOTPANELS=*n*	specifies number of pages or screens to be used to display chart
ZEROSTD	displays ANOM chart regardless of whether the root mean square error is zero

Table 2.14. Graphical Enhancement Options

ANNOTATE= *SAS-data-set*	specifies annotate data set that adds features to ANOM chart
DESCRIPTION='*string*'	specifies string that appears in the description field of the PROC GREPLAY master menu for ANOM chart
FONT=*font*	specifies software font for labels and legends on charts
NAME='*string*'	specifies name that appears in the name field of the PROC GREPLAY master menu for ANOM chart
PAGENUM='*string*'	specifies the form of the label used in pagination
PAGENUMPOS= *keyword*	specifies the position of the page number requested with the PAGENUM= option

Table 2.15. Star Options

CSTARCIRCLES=*color*	specifies color for STARCIRCLES= circles
CSTARFILL=*color*| *(variable)*	specifies color for filling stars
CSTAROUT=*color*	specifies outline color for stars exceeding inner or outer circles
CSTARS=*color*| *(variable)*	specifies color for outlines of stars
LSTARCIRCLES= *linetypes*	specifies line types for STARCIRCLES= circles
LSTARS=*linetype*|*(variable)*	specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=*value*	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=*value-list*	specifies reference circles for stars
STARINRADIUS=*value*	specifies inner radius of stars
STARLABEL=*keyword*	specifies vertices to be labeled
STARLEGEND=*keyword*	specifies style of legend for star vertices
STARLEGENDLAB=*'label'*	specifies label for STARLEGEND= legend
STAROUTRADIUS=*value*	specifies outer radius of stars
STARSPEC=*value*| *SAS-data-set*	specifies method used to standardize vertex variables
STARSTART=*value*	specifies angle for first vertex
STARTYPE=*keyword*	specifies graphical style of star
STARVERTICES= *variable*|*(variables)*	superimposes star at each point on ANOM chart
WSTARCIRCLES=*n*	specifies width of STARCIRCLES= circles
WSTARS=*n*	specifies width of STARVERTICES= stars

Table 2.16. Overlay Options

CCOVERLAY=(*color-list*)	specifies colors for ANOM chart overlay line segments
COVERLAY=(*color-list*)	specifies colors for ANOM chart overlay plots
COVERLAYCLIP=*color*	specifies color for clipped points on overlays
LOVERLAY=(*linetypes*)	specifies line types for ANOM chart overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=(*variable-list*)	specifies variables to overlay on ANOM chart
OVERLAYCLIPSYM=*symbol*	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=*value*	specifies symbol height for clipped points on overlays
OVERLAYHTML=(*variable-list*)	specifies URLS to associate with ANOM chart overlay points
OVERLAYLEGLAB='*label*'	specifies label for overlay legend
OVERLAYSYM=(*symbol-list*)	specifies symbols for ANOM chart overlays
OVERLAYSYMHT=(*value-list*)	specifies symbol heights for ANOM chart overlays
WOVERLAY=(*value-list*)	specifies widths of ANOM chart overlay line segments

Details

Constructing ANOM Charts for Means

The following notation is used in this section:

X_{ij}	jth response in the ith group
k	number of groups
n_i	sample size of ith group
N	total sample size $= n_1 + \cdots + n_k$
μ_i	expected value of the responses in the ith group
σ	standard deviation of the responses in the ith group
$\bar{X}_i$	mean of responses in ith group
$\overline{\overline{X}}$	weighted average of k group means
s_i^2	variance of responses in ith group
$\widehat{\sigma^2}$	mean square error (MSE)
ν	degrees of freedom associated with the mean square error
α	significance level
$h(\alpha; k, n, \nu)$	critical value for analysis of means when the sample sizes n_i are equal $(n_i \equiv n)$
$h(\alpha; k, n_1, \ldots, n_k, \nu)$	critical value for analysis of means when the sample sizes n_i are not equal

Plotted Points

Each point on an ANOM chart indicates the value of a group mean ($\bar{X}_i$).

Central Line

By default, the central line on an ANOM chart for means represents the weighted average of the group means, which is computed as

$$\overline{\overline{X}} = \frac{n_1\bar{X}_1 + \cdots + n_k\bar{X}_k}{n_1 + \cdots + n_k}$$

You can specify a value for $\overline{\overline{X}}$ with the MEAN= option in the XCHART statement or with the variable _MEAN_ in a LIMITS= data set.

Decision Limits

In the analysis of means for continuous data, it is assumed that the responses in the ith group are at least approximately normally distributed with a constant variance:

$$X_{ij} \sim N(\mu_i, \sigma^2), \quad j = 1, \ldots, n_i$$

When the group sizes are constant ($n_i \equiv n$), then $\nu = N - k = k(n-1)$ and the decision limits are computed as follows:

$$\begin{aligned} \text{lower decision limit (LDL)} &= \overline{\overline{X}} - h(\alpha; k, n, \nu)\sqrt{\text{MSE}}\sqrt{\frac{k-1}{N}} \\ \text{upper decision limit (UDL)} &= \overline{\overline{X}} + h(\alpha; k, n, \nu)\sqrt{\text{MSE}}\sqrt{\frac{k-1}{N}} \end{aligned}$$

Here the mean square error (MSE) is computed as follows:

$$\text{MSE} = \widehat{\sigma^2} = \frac{1}{k}\sum_{j=1}^{k} s_j^2$$

For details concerning the function $h(\alpha; k, n, \nu)$, see Nelson (1982, 1993).

When the group sizes are not constant (the unbalanced case), $\nu = N - k$ and the decision limits for the ith group are computed as follows:

$$\begin{aligned} \text{lower decision limit (LDL)} &= \overline{\overline{X}} - h(\alpha; k, n_1, \ldots, n_k, \nu)\sqrt{\text{MSE}}\sqrt{\frac{N-n_i}{Nn_i}} \\ \text{upper decision limit (UDL)} &= \overline{\overline{X}} + h(\alpha; k, n_1, \ldots, n_k, \nu)\sqrt{\text{MSE}}\sqrt{\frac{N-n_i}{Nn_i}} \end{aligned}$$

Here the mean square error (MSE) is computed as follows:

$$\text{MSE} = \widehat{\sigma^2} = \frac{(n_1-1)s_1^2 + \cdots + (n_k-1)s_k^2}{n_1 + \cdots + n_k - k}$$

This requires that ν be positive. A chart is not produced if $\nu > 0$ but MSE is equal to zero (unless you specify the ZEROSTD option). For details concerning the function $h(\alpha; k, n_1, \ldots, n_k, \nu)$, see Nelson (1991).

You can specify parameters for the limits as follows:

- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set. By default, α = 0.05.
- Specify a constant nominal sample size $n_i \equiv n$ for the decision limits in the balanced case with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set. By default, n is the observed sample size in the balanced case.
- Specify k with the LIMITK= option or with the variable _LIMITK_ in a LIMITS= data set. By default, k is the number of groups.

- Specify $\overline{\overline{X}}$ with the MEAN= option or with the variable _MEAN_ in a LIMITS= data set. By default, $\overline{\overline{X}}$ is the weighted average of the responses.
- Specify $\widehat{\sigma^2}$ with the MSE= option or with the variable _MSE_ in a LIMITS= data set. By default, $\widehat{\sigma^2}$ is computed as indicated above.
- Specify ν with the DFE= option or with the variable _DFE_ in a LIMITS= data set. By default, ν is determined as indicated above.

Constructing ANOM Charts for Two-Way Layouts

This section provides the computational details for constructing an ANOM chart for the lth factor in an experiment involving two factors (l = 1 or 2). It is assumed that there is no interaction effect. See Example 2.2 for an illustration.

The following notation is used in this section:

X_{ijk}	kth response at the ith level of factor 1 and the jth level of factor 2, where $k = 1, 2, \ldots, n_{ij}$
f_l	number of groups (levels) for the lth factor, $l = 1, 2$
n_{ij}	number of replicates in cell (i, j)
N	total sample size $= \sum_{i=1}^{f_1} \sum_{j=1}^{f_2} n_{ij}$
σ^2	variance of the responses
$\overline{X}_{ij\cdot}$	average response in cell (i, j)
$\overline{X}_{i\cdot\cdot}$	average response for ith level of factor 1 $= \left(\sum_{j=1}^{f_2} n_{ij}\overline{X}_{ij\cdot}\right) / \left(\sum_{j=1}^{f_2} n_{ij}\right)$
$\overline{X}_{\cdot j\cdot}$	average response for jth level of factor 2 $= \left(\sum_{i=1}^{f_1} n_{ij}\overline{X}_{ij\cdot}\right) / \left(\sum_{i=1}^{f_1} n_{ij}\right)$
$\overline{\overline{X}}$	$\sum_{i=1}^{f_1} \sum_{j=1}^{f_2} n_{ij}\overline{X}_{ij}/N$
s_{ij}^2	variance of responses in the ith level of factor 1 and the jth level of factor 2
$\widehat{\sigma^2}$	mean square error (MSE) in the two-way analysis of variance
ν	degrees of freedom associated with the mean square error in the two-way analysis of variance
α	significance level
$h(\alpha; f_l, n, \nu)$	critical value for analysis of means in a one-way layout for f_l groups (treatment levels) when the sample sizes for each level are constant ($\equiv n$) and ν is the degrees of freedom associated with the mean square error; see page 31.

Plotted Points

The points on the ANOM chart for factor 1 represent $\overline{X}_{i\cdot\cdot}$, $i = 1, \ldots, f_1$ and the points on the ANOM chart for factor 2 represent $\overline{X}_{\cdot j\cdot}$, $j = 1, \ldots, f_2$.

Central Line

The central line on the ANOM chart for the lth factor is the overall weighted average $\overline{\overline{X}}$. Some authors use the notation $\overline{X}_{...}$ for this average.

Decision Limits

It is assumed that

$$X_{ijk} = \mu + \alpha_i + \beta_j + \epsilon_{ijk}$$

where the quantities ϵ_{ijk} are independent and at least approximately normally distributed with

$$\epsilon_{ijk} \sim N(0, \sigma^2)$$

The correct decision limits for a given factor in a two-way layout are not computed by default when the lth factor is specified as the *group-variable* in the XCHART statement, since the mean square error and degrees of freedom are not adjusted for the two-way structure of the data. Consequently, $\widehat{\sigma^2}$ and ν must be precomputed and provided to the ANOM procedure, as illustrated in Example 2.2.

In the case of a two-way layout with equal group sizes ($n_{ij} \equiv n$), the appropriate decision limits are:

$$\begin{aligned} \text{lower decision limit (LDL)} &= \overline{\overline{X}} - h(\alpha; f_l, n, \nu)\sqrt{\text{MSE}}\sqrt{\frac{f_l - 1}{N}} \\ \text{upper decision limit (UDL)} &= \overline{\overline{X}} + h(\alpha; f_l, n, \nu)\sqrt{\text{MSE}}\sqrt{\frac{f_l - 1}{N}} \end{aligned}$$

where the mean square error (MSE) is computed as in the ANOVA or GLM procedure:

$$\text{MSE} = \widehat{\sigma^2} = \frac{1}{f_1 f_2} \sum_{i=1}^{f_1} \sum_{j=1}^{f_2} s_{ij}^2$$

and the degrees of freedom for error is $\nu = f_1 f_2 (n-1)$. For details concerning the function $h(\alpha; f_l, n, \nu)$, see Nelson (1982a, 1993).

You can provide the appropriate values of MSE and ν by

- specifying $\widehat{\sigma^2}$ with the MSE= option or with the variable _MSE_ in a LIMITS= data set
- specifying ν with the DFE= option or with the variable _DFE_ in a LIMITS= data set

In addition you can:

- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set. By default, $\alpha = 0.05$.
- Specify a constant nominal sample size $n_{ij} \equiv n$ for the decision limits in the balanced case with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify f_l with the LIMITK= option or with the variable _LIMITK_ in a LIMITS= data set.
- Specify $\overline{\overline{X}}$ with the MEAN= option or with the variable _MEAN_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves decision limits and decision limit parameters. The following variables can be saved:

Table 2.17. OUTLIMITS= Data Set

Variable	Description
ALPHA	significance level (α).
DFE	degrees of freedom for mean square error (ν).
GROUP	*group-variable* specified in the XCHART statement
INDEX	optional identifier for the decision limits specified with the OUTINDEX= option
LDLX	lower decision limit for group means
LIMITK	number of groups
LIMITN	group sample size associated with the decision limits
MEAN	weighted average of group means ($\overline{\overline{X}}$)
MSE	mean square error ($\widehat{\sigma^2}$).
TYPE	type (estimate or standard value) of _MEAN_ and _MSE_
UDLX	upper decision limit for group means
VAR	*response* specified in the XCHART statement

Notes:

1. In the unbalanced case, the special missing value *V* is assigned to the variables _LIMITN_, _LDLX_, and _UDLX_ to indicate that the decision limits vary with the group sample size.
2. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *response* specified in the XCHART statement. For an example, see the section "Saving Decision Limits" on page 21.

OUTSUMMARY= Data Set

The OUTSUMMARY= data set saves group summary statistics. The following variables can be saved:

- the *group-variable*
- a group mean variable named by *response* suffixed with *X*
- a group sample size variable named by *response* suffixed with *N*
- a group standard deviation variable named by *response* suffixed with *S*

Given a *response* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Group summary variables are created for each *response* specified in the XCHART statement. For example, consider the following statements:

```
proc anom data=Steel;
   xchart (Width Diameter)*lot / outsummary=Summary;
run;
```

The data set Summary contains variables named lot, WidthX, WidthS, WidthN, DiameterX, DiameterS, and DiameterN. Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTSUMMARY= data set, see the section "Saving Summary Statistics for Groups" on page 20.

OUTTABLE= Data Set

The OUTTABLE= data set saves group summary statistics, decision limits, and related information. The following variables can be saved:

Variable	Description
ALPHA	significance level (α)
EXLIM	decision limit exceeded (if any)
group	values of the group variable
LDLX	lower decision limit for group mean
LIMITN	nominal sample size associated with the decision limits
MEAN	central line
SUBN	group sample size
SUBX	group mean
UDLX	upper decision limit for group mean
VAR	*response* specified in the XCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Note: The variable _EXLIM_ is a character variable of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see the section "Saving Decision Limits" on page 21.

ODS Tables

The following table summarizes the ODS tables that you can request with the XCHART statement.

Table 2.18. ODS Tables Produced with the XCHART Statement

Table Name	Description	Options
XCHART	ANOM chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLEOUT,

Input Data Sets

DATA= Data Set

You can read raw data (response values) from a DATA= data set specified in the PROC ANOM statement. Each *response* specified in the XCHART statement must be a SAS variable in the DATA= data set. This variable provides measurements that must be grouped into group samples indexed by the *group-variable*. The *group-variable*, which is specified in the XCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *response* and a value for the *group-variable*. If the ith group contains n_i items, there should be n_i consecutive observations for which the value of the *group-variable* is the index of the ith group. For example, if each group contains five items and there are 10 groups, the DATA= data set should contain 50 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the ANOM procedure reads all of the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option.

For an example of a DATA= data set, see the section "Creating ANOM Charts for Means from Response Values" on page 15.

LIMITS= Data Set

You can read preestablished decision limits (or parameters from which the decision limits can be calculated) from a LIMITS= data set specified in the PROC ANOM statement. For example, the following statements read decision limit information from the data set conlims:

```
proc anom data=info limits=conlims;
   xchart weight*batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the ANOM procedure. Such data sets always contain the variables required for a LIMITS= data set; see Table 2.17 on page 35. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following minimal combinations of variables:

- the variables _LDLX_, _MEAN_, and _UDLX_, which specify the decision limits directly
- the variables _MEAN_ and _MSE_, with _DFE_ recommended, which are used to calculate the decision limits according to the equations in the section "Decision Limits" on page 31

In addition, note the following:

- The variables _VAR_ and _GROUP_ are always required. These must be character variables whose lengths are no greater than 32.
- _DFE_ is optional. The default is $\nu = N - k$, and in the case of equal group sizes, $\nu = k(n-1)$.
- _MSE_ is optional if _LDLX_ and _UDLX_ are specified; otherwise it is required.
- _LDLX_ and _UDLX_ must be specified together; otherwise their values are computed.
- _ALPHA_ is optional but is recommended in order to maintain a complete set of decision limit information. The default value is 0.05.
- _LIMITK_ is optional. The default value is k, the number of groups. A group must have at least one non-missing value ($n_i \geq 1$) and there must be at least one group with $n_i \geq 2$. If specified, _LIMITK_ overrides the value of k.

- _LIMITN_ is optional. The default value is the common group size (n), in the balanced case $n_i \equiv n$. If specified, _LIMITN_ overrides the value of n.
- The variable _TYPE_ is optional, but is recommended to maintain a complete set of decision limit information. The variable _TYPE_ must be a character variable of length 8. Valid values are `ESTIMATE`, `STANDARD`, `STDMEAN`, and `STDRMS`. The default is `ESTIMATE`.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- BY variables are required if specified with a BY statement.

SUMMARY= Data Set

You can read group summary statistics from a SUMMARY= data set specified in the PROC ANOM statement. This enables you to reuse OUTSUMMARY= data sets that have been created in previous runs of the ANOM procedure or to read output data sets created with SAS summarization procedures, such as PROC MEANS.

A SUMMARY= data set used with the XCHART statement must contain the following:

- the *group-variable*
- a group mean variable for each *response*
- a group sample size variable for each *response*
- a group standard deviation variable for each *response*

The names of the group mean, group range, and group sample size variables must be the *response* name concatenated with the suffix characters X, S, and N, respectively.

For example, consider the following statements:

```
proc anom summary=Summary;
   xchart (Weight Yldstren)*batch;
run;
```

The data set Summary must include the variables batch, WeightX, WeightS, WeightN, YldsrenX, YldsrenS, and YldsrenN. Note that if you specify a *response* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *response* name, suffixed with the appropriate character.

Other variables that can be read from a SUMMARY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the ANOM procedure reads all of the observations in a SUMMARY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option.

For an example of a SUMMARY= data set, see the section "Creating ANOM Charts for Means from Group Summary Data" on page 18.

TABLE= Data Set

You can read summary statistics and decision limits from a TABLE= data set specified in the PROC ANOM statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the ANOM procedure. Because the ANOM procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized ANOM charts.

The following table lists the variables required in a TABLE= data set used with the XCHART statement:

Table 2.19. Variables Required in a TABLE= Data Set

Variable	Description
group-variable	values of the *group-variable*
LDLX	lower decision limit for mean
LIMITN	nominal sample size associated with the decision limits
MEAN	central line
SUBN	group sample size
SUBX	group mean
UDLX	upper decision limit for mean

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _VAR_. This variable is required if more than one *response* is specified or if the data set contains information for more than one *response*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see the section "Saving Decision Limits" on page 21.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*group-variable*
Vertical	DATA=	*response*
Vertical	SUMMARY=	group mean variable
Vertical	TABLE=	_SUBX_

Missing Values

An observation read from a DATA=, SUMMARY=, or TABLE= data set is not analyzed if the value of the group variable is missing. For a particular response variable, an observation read from a DATA= data set is not analyzed if the value of the response variable is missing. Missing values of response variables generally lead to unequal group sample sizes. For a particular response variable, an observation read from a SUMMARY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples

This section provides advanced examples of the XCHART statement.

Example 2.1. ANOM Charts with Unequal Group Sizes

See ANMXEX1 in the SAS/QC Sample Library

Consider the example described in "Creating ANOM Charts for Means from Response Values" on page 15. Suppose that four of the 10 measurements were missing for the third and fourth labeler positions. The following statements create a SAS data set named LabelDev2, which contains the resulting deviation measurements:

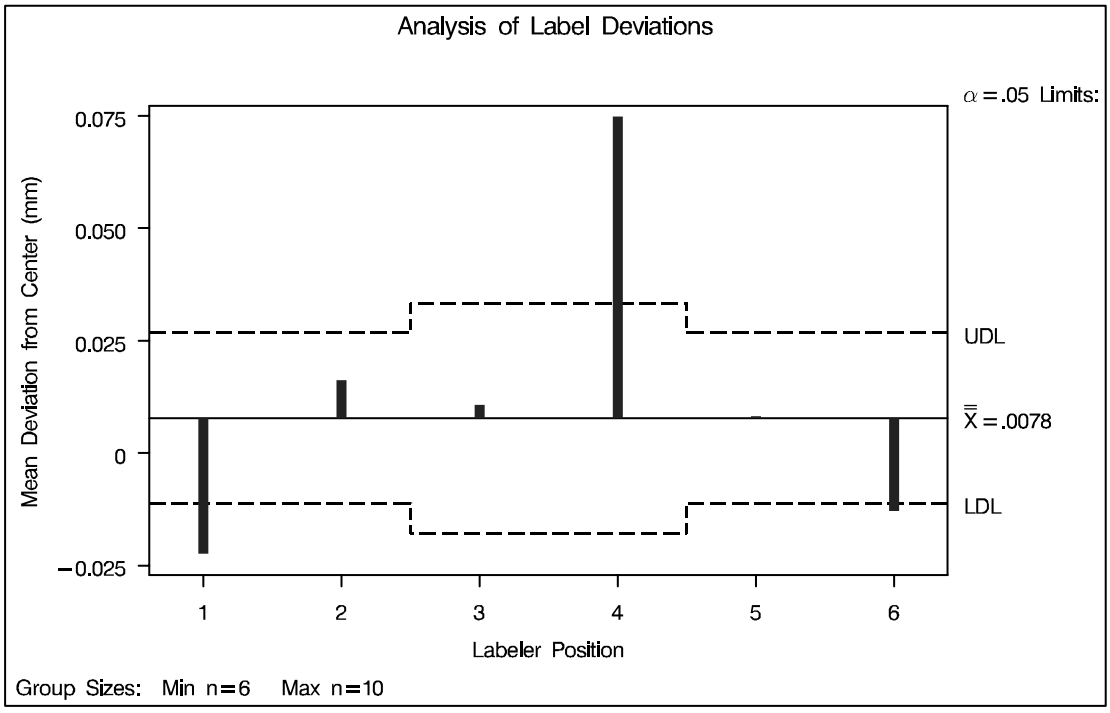
```
data LabelDev2;
   input Position @;
   do i = 1 to 5;
      input Deviation @;
      output;
   end;
   drop i;
   datalines;
1  -0.0239  -0.0285  -0.0300  -0.0043  -0.0362
1  -0.0422  -0.0014  -0.0647   0.0094  -0.0016
2  -0.0201  -0.0273   0.0227  -0.0332   0.0366
2   0.0438   0.0556   0.0098   0.0564   0.0182
3  -0.0073   0.0285    .        .      -0.0139
3    .       0.0357   0.0235   .      -0.0020
4   0.0669   0.1073    .        .       0.0755
4    .       0.0561   0.0899   .       0.0530
```

```
5    0.0368    0.0036    0.0374    0.0116   -0.0074
5    0.0250   -0.0080    0.0302   -0.0015   -0.0464
6    0.0049   -0.0384   -0.0204   -0.0049   -0.0120
6    0.0071   -0.0308    0.0017   -0.0285   -0.0070
run;
```

The following statements create the ANOM chart shown in Output 2.1.1:

```
title 'Analysis of Label Deviations';
proc anom data=LabelDev2;
   xchart Deviation*Position;
   label Deviation = 'Mean Deviation from Center (mm)';
   label Position  = 'Labeler Position';
run;
```

Output 2.1.1. ANOM Chart with Unequal Group Sizes



Note that the decision limits are automatically adjusted for the varying group sizes. The legend reports the minimum and maximum group sizes.

Example 2.2. ANOM for a Two-Way Classification

See ANMXEX2 in the SAS/QC Sample Library

A chemical engineer is interested in the effects of two factors, position and depth, on the concentration of a cleaning solution; refer to Ramig (1983) for details concerning the use of ANOM in a two-way classification such as this. The engineer is interested in the following questions:

1. Are there significant group or interaction effects due to position or depth?
2. Assuming a main effect is significant, which levels are significantly different from the overall mean and in which direction?

There are five positions and three depths. The engineer runs a two-way factorial experiment with two replications per cell. The following statements create a data set named Cleaning, which provides the concentration measurements for the $5 \times 3 \times 2 = 30$ observations.

```
data cleaning;
   do position = 1 to 5;
      do depth = 1 to 3;
         do rep = 1 to 2;
            input concentration @@;
            output;
         end;
      end;
   end;
   datalines;
   15 16 15 14 19 5
   15 16 14 14 0  8
   19 15 16 16 11 8
   18 16 19 15 8  14
   15 12 19 15 8  11
   ;
run;
```

In order to test for main effects and an interaction effect, the following statements use the GLM procedure:

```
proc glm data=cleaning;
   class position depth;
   model concentration = position depth position*depth;
run;
```

The results are shown in Output 2.2.1:

Output 2.2.1. GLM Results

The GLM Procedure

Dependent Variable: concentration

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	14	390.4666667	27.8904762	2.21	0.0694
Error	15	189.0000000	12.6000000		
Corrected Total	29	579.4666667			

R-Square	Coeff Var	Root MSE	concentration Mean
0.673838	26.22893	3.549648	13.53333

Source	DF	Type I SS	Mean Square	F Value	Pr > F
position	4	50.4666667	12.6166667	1.00	0.4374
depth	2	281.6666667	140.8333333	11.18	0.0011
position*depth	8	58.3333333	7.2916667	0.58	0.7802

Source	DF	Type III SS	Mean Square	F Value	Pr > F
position	4	50.4666667	12.6166667	1.00	0.4374
depth	2	281.6666667	140.8333333	11.18	0.0011
position*depth	8	58.3333333	7.2916667	0.58	0.7802

The results in Output 2.2.1 show no significant interaction effect* and a significant main effect due to depth. Since no interaction effect is present, you can use analysis of means to evaluate the effect of each factor as if two separate experiments had been run to determine the effect of each factor. In other words, the analysis of means is done twice, once for each factor. However, each analysis must be based on the mean square error ($\widehat{\sigma^2} = 12.6$) and the degrees of freedom for error ($\nu = 15$) from the two-way analysis of variance. These values must be specified since the ANOM procedure assumes a one-way layout by default for computing the decision limits.

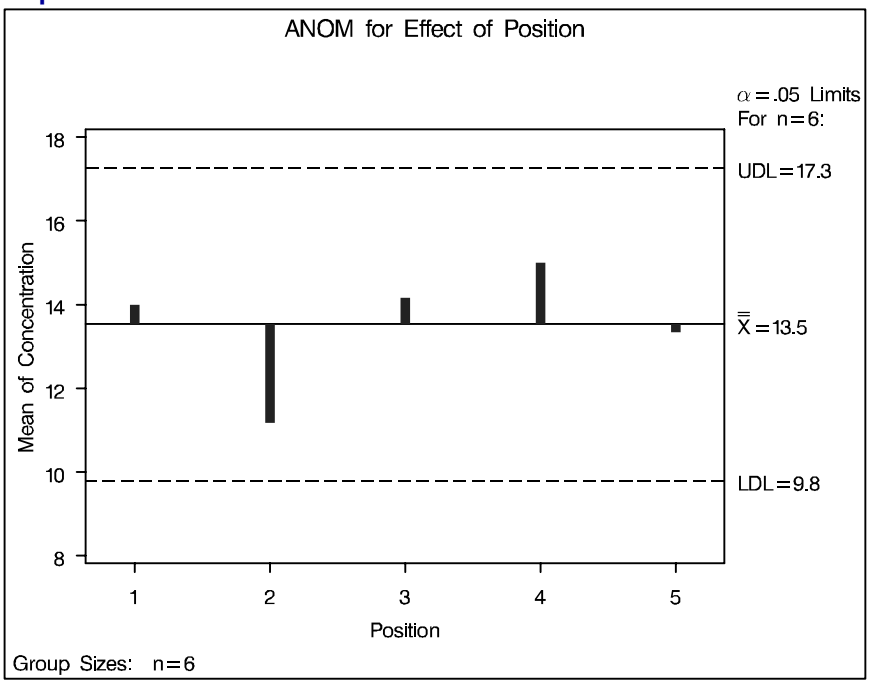
The following statements create the ANOM chart for the effect of Position shown in Output 2.2.2:

```
title "ANOM for Effect of Position";
proc anom data=cleaning;
   xchart concentration * position / mse      = 12.6
                                    dfe      = 15
                                    outtable = posmain;
   label position      = 'Position'
         concentration = 'Mean of Concentration';
run;
```

*See Example 2.4 for an example that discusses the use of ANOM for the cell means when an interaction effect is present.

The MSE= and DFE= options are used to specify $\widehat{\sigma^2}$ and ν respectively. See the section "Constructing ANOM Charts for Two-Way Layouts" on page 33 for how the specified values are used to compute the decision limits. The OUTTABLE= option stores the output data set PosMain, which can be used to create a combined chart for the two factors.

Output 2.2.2. ANOM for Effect of Position



Each point on the ANOM chart represents the average response for a particular level of position. In this case, all of the points are between the upper decision limit (UDL) and the lower decision limit (LDL). This is not surprising considering the fact that the main effect of Position was not significant in the ANOVA.

In order to create the ANOM chart for the effect of depth, the response must be sorted by Depth.

```
proc sort data=cleaning out=cleaning2;
   by depth;
run;
```

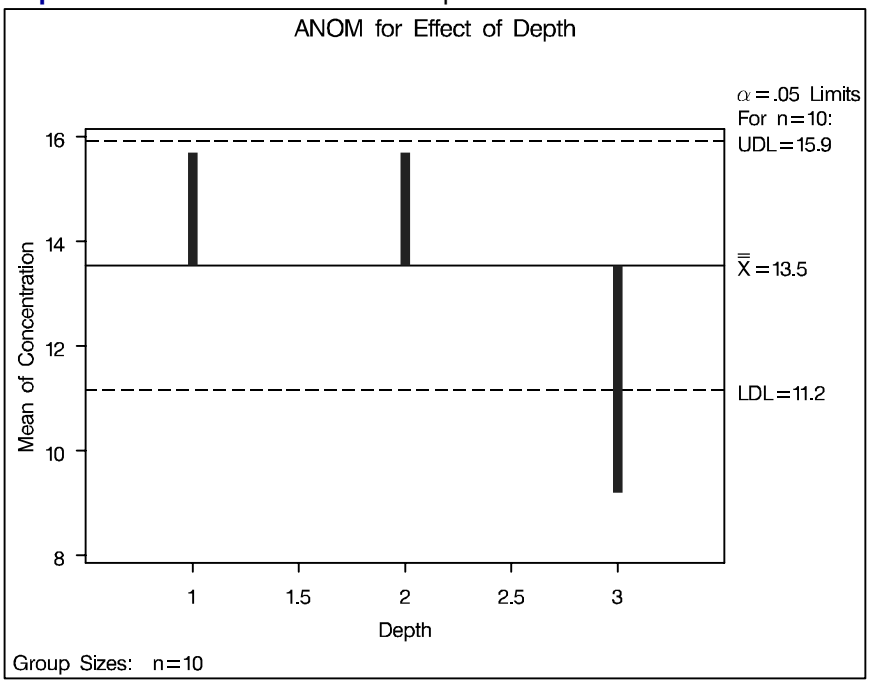
Note that for the previous chart, the measurements were sorted by Position in the original data set.

The following statements create the chart for Depth:

```
title "ANOM for Effect of Depth";
proc anom data=cleaning2;
   xchart concentration * depth / mse      = 12.6
                                  dfe      = 15
                                  outtable = depmain;
   label depth           = 'Depth'
         concentration   = 'Mean of Concentration';
run;
```

This produces the chart shown in Output 2.2.3: The OUTTABLE= option stores the output data set DepMain, which can be used to create a combined chart for the two factors.

Output 2.2.3. ANOM for Effect of Depth



Since the average concentration for Depth 3 is less than the lower decision limit, you can conclude that the average response for Depth 3 is significantly less than the overall mean.

Example 2.3. ANOM Charts Using LIMITS= Data Set

See ANMXEX4 in the SAS/QC Sample Library

In Example 2.2, statistics from a two-way ANOVA were passed to the ANOM procedure using options in order to compute the decision limits for the factor effects. This example shows how you can pass the statistics in a LIMITS= data set using the variables _MSE_ and _DFE_.

The GLM output in Output 2.2.1 on page 44 provides the statistics. The following statements save the results from PROC GLM in the data sets MyFit, MyMeans, and MyOverAll:

```
ods select FitStatistics ModelANOVA OverAllANOVA;
ods output FitStatistics = MyFit
           ModelANOVA    = MyLimits
           OverAllANOVA  = MyOverAll;

proc glm data=cleaning;
   class position depth;
   model concentration = position depth position*depth;
run;
```

The results of PROC GLM are identical to the results in Output 2.2.1.

The following statements create a LIMITS= data set to be used to create an ANOM chart for the effect of Position:

```
data ANOMParms;
   keep _var_ _group_ _alpha_ _mean_;
      length _var_ _group_ $ 14;
   set MyFit (rename=(Dependent=_var_ DepMean  =_mean_));
      _group_ = 'position';
      _alpha_ = 0.05;

data ANOMParms;
   merge ANOMParms
         MyLimits (where=(source='position')
                   keep = source DF);
                   _limitk_ = DF+1;
   drop source DF;
   merge MyOverAll (where=(source='Error')
            keep = source df ms
            rename=( df = _dfe_ ms = _mse_));
   drop source;
   merge MyOverAll (where=(source='Corrected Total')
                    keep = source DF);
                    _limitn_ = (DF+1)/_limitk_;
   drop source DF;
run;
```

The data set ANOMParms contains a complete set of parameters, as shown in Output 2.3.1. Note these are the same values specified in the options for Example 2.2.

Output 2.3.1. Data Set ANOMParms

Parameters for ANOM for Effect of Position

var	_group_	_mean_	_alpha_	_limitk_	_dfe_	_mse_	_limitn_
concentration	position	13.53333	0.05	5	15	12.6000000	6

The following statements read the parameters in ANOMParms to create an ANOM chart for the effect of Position:

```
title "ANOM for Effect of Position";
proc anom data=cleaning limits=ANOMParms;
   xchart concentration * position /outtable = postable;
   label position      = 'Position'
         concentration = 'Mean of Concentration';
run;
```

The chart produced is identical to the one in Output 2.2.2. Note that the procedure creates a TABLE= input data set PosTable. You can use PosTable to create a combined chart for the two factors Position and Depth.

You can create a LIMITS= data set ANOMParmsB for the factor Depth by using the above code and substituting 'depth' for the _group_ variable. You can use the OUTTABLE= statement to store the TABLE= input data set for Depth as DepTable. The resulting data set ANOMParmsB is shown in Output 2.3.2:

Output 2.3.2. Data Set ANOMParmsB

Parameters for ANOM for Effect of Depth

var	_group_	_mean_	_alpha_	_limitk_	_dfe_	_mse_	_limitn_
concentration	depth	13.53333	0.05	3	15	12.6000000	10

Example 2.4. ANOM for Cell Means in Presence of Interaction

See ANMXEX6 in the SAS/QC Sample Library

This example illustrates the use of analysis of means in an experiment with two factors where an interaction effect is present. The following data set CleaningInteract is a modified version of the data set Cleaning, which includes an interaction effect for Position and Depth.

Consider the following data set CleaningInteract:

```
data cleaninginteract;

   do position = 1 to 5;
      do depth = 1 to 3;
         do rep = 1 to 2;
            input concentration @@;
            output;
         end;
      end;
   end;
   datalines;
   15 16 15 14 19 5
   15 16 14 14 0  1
   19 15 16 16 11 8
   18 16 24 23 8  14
   15 12 23 24 8  11
   ;
run;
```

The following statements use PROC GLM to test for an interaction:

```
proc glm data=cleaninginteract;
   class position depth;
   model concentration = position depth position*depth;
run;
```

The analysis of variance results in Output 2.4.1 indicate a significant interaction between Position and Depth.

Output 2.4.1. GLM Results

The GLM Procedure

Dependent Variable: concentration

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	14	885.666667	63.261905	6.66	0.0004
Error	15	142.500000	9.500000		
Corrected Total	29	1028.166667			

R-Square	Coeff Var	Root MSE	concentration Mean
0.861404	21.75676	3.082207	14.16667

Source	DF	Type I SS	Mean Square	F Value	Pr > F
position	4	169.0000000	42.2500000	4.45	0.0144
depth	2	515.4666667	257.7333333	27.13	<.0001
position*depth	8	201.2000000	25.1500000	2.65	0.0496

Source	DF	Type III SS	Mean Square	F Value	Pr > F
position	4	169.0000000	42.2500000	4.45	0.0144
depth	2	515.4666667	257.7333333	27.13	<.0001
position*depth	8	201.2000000	25.1500000	2.65	0.0496

Since an interaction effect is present, an appropriate way to analyze the data is to create an ANOM chart for the cell means.

In order to create the chart you first need to compute the cell means and a new *group* variable which designates the cells. The following statements use PROC MEANS for this purpose.

```
proc means data=cleaninginteract n mean std;
   class position depth;
   var concentration;
   types position*depth;
   output out=cellmeans mean=concentrationX std=concentrationS;
run;

data cellmeans; set cellmeans;
   rename _FREQ_ = concentrationN;
   pos = put(position, z1.);
   dep = put(depth, z1.);
   cell  = cat('P',pos, 'D', dep);
   drop _TYPE_ pos dep;
run;
```

The cell means are stored in the data set CellMeans shown in Output 2.4.2:

Output 2.4.2. Data Set CellMeans

position	depth	concentration N	concentration X	concentration S	cell
1	1	2	15.5	0.70711	P1D1
1	2	2	14.5	0.70711	P1D2
1	3	2	12.0	9.89949	P1D3
2	1	2	15.5	0.70711	P2D1
2	2	2	14.0	0.00000	P2D2
2	3	2	0.5	0.70711	P2D3
3	1	2	17.0	2.82843	P3D1
3	2	2	16.0	0.00000	P3D2
3	3	2	9.5	2.12132	P3D3
4	1	2	17.0	1.41421	P4D1
4	2	2	23.5	0.70711	P4D2
4	3	2	11.0	4.24264	P4D3
5	1	2	13.5	2.12132	P5D1
5	2	2	23.5	0.70711	P5D2
5	3	2	9.5	2.12132	P5D3

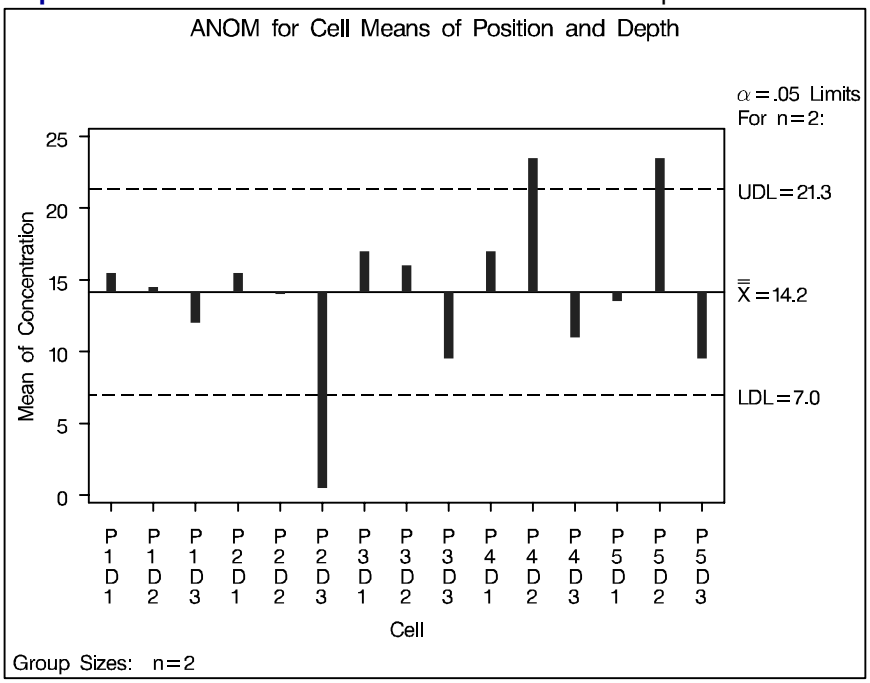
The data set CellMeans has the structure of a SUMMARY= input data set for the ANOM procedure. For details concerning a SUMMARY= data set, see the section "Creating ANOM Charts for Means from Group Summary Data" on page 18.

The following statements use CellMeans to create the ANOM chart for the cell means using SUMMARY= option:

```
title "ANOM for Cell Means of Position and Depth";
proc ANOM summary = cellmeans;
   xchart concentration * cell / turnhlabels;
   label concentrationX = 'Mean of Concentration';
   label cell           = 'Cell';
run;
```

The chart is shown in Output 2.4.3:

Output 2.4.3. ANOM for Cell Means of Position and Depth



The chart shows that the cell means for P2D3, P4D2, and P5D2 are significantly different from the average concentration level.

Chapter 3
PCHART Statement

Chapter Contents

OVERVIEW 55

GETTING STARTED 55
- Creating ANOM Charts for Proportions from Group Counts 55
- Creating ANOM Charts for Proportions from Group Summary Data 58
- Saving Group Proportions 60
- Saving Decision Limits 61

SYNTAX 63
- Summary of Options 64

DETAILS 72
- Constructing ANOM Charts for Proportions 72
- Output Data Sets 74
- ODS Tables 76
- Input Data Sets 76
- Axis Labels 80
- Missing Values 80

EXAMPLES 81
- Example 3.1. ANOM p Charts with Angled Axis Labels 81

Chapter 3
PCHART Statement

Overview

The PCHART statement creates ANOM charts for group (treatment level) proportions, also referred to as ANOM p *charts*.

You can use options in the PCHART statement to

- compute decision limits from the data based on specified parameters, such as the significance level (α)
- tabulate group sample sizes, group proportions, decision limits, and other information
- save decision limits in an output data set
- save group sample sizes and group proportions in an output data set
- read decision limits and decision limit parameters from a data set
- display distinct sets of decision limits for different sets of groups on the same chart
- add block legends and symbol markers to identify special groups
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

Getting Started

This section introduces the PCHART statement with simple examples that illustrate commonly used options. Complete syntax for the PCHART statement is presented in the "Syntax" section on page 63.

Creating ANOM Charts for Proportions from Group Counts

A health care system administrator uses ANOM to compare cesarean section rates for a set of medical groups. For more background concerning this application, refer to Rodriguez (1996).

See ANMP1 in the SAS/QC Sample Library

The following statements create a SAS data set named Csection, which contains the number of c-sections and the total number of deliveries for each medical group over a one-year period.

```
data Csection;
   length ID $ 2;
   input ID Csections Total @@;
   label ID = 'Medical Group Identification Number';
   datalines;
1A  150  923 1K   45  298 1B   34  170 1D   18  132
3I   20  106 3M   12  105 1E   10   77 1N   19   74
1Q    7   69 3H   11   65 1R   11   49 1H    9   48
3J    7   20 1C    8   43 3B    6   43 1M    4   29
3C    5   28 1O    4   27 1J    6   22 1T    3   22
3E    4   18 1G    4   15 3D    4   13 3G    1   11
1L    2   10 1I    1    8 1P    0    3 1F    0    3
1S    1    3
;
run;
```

A partial listing of CSECTION is shown in Figure 3.1.

Cesarean Section Data

ID	Csections	Total
1A	150	923
1K	45	298
1B	34	170
1D	18	132
3I	20	106
3M	12	105
1E	10	77
1N	19	74
1Q	7	69
3H	11	65

Figure 3.1. The Data Set Csection

The variable ID identifies the medical groups and is referred to as the *group-variable*. The variable Csections provides the number of c-sections, and is referred to as the *response variable* (or *response* for short). The variable Total provides the total number of deliveries.

The following statements create the p chart shown in Figure 3.2:

```
title 'Analysis of C-Sections';
proc anom data=Csection;
   pchart Csections*ID / groupn   = Total
                         nolegend
                         turnhlabels;
   label Csections = 'Proportion of Cesarean Sections';
run;
```

This example illustrates the basic form of the PCHART statement. After the keyword PCHART, you specify the *response* to analyze (in this case, Csections, followed by an asterisk and the *group-variable* ID.

The input data set is specified with the DATA= option in the PROC ANOM statement. The GROUPN= option specifies the sample size in each group and is required with a DATA= input data set. The GROUPN= option specifies one of the following:

- a constant group sample size
- a variable in the input data set whose values provide the group sample sizes (in this case, Total)

The TURNHLABELS option turns the horizontal axis labels since the default labeling skips labels if the characters exceed the space alloted. See Axis and Axis Label Options on page 66. To angle the axis labels, see Example 3.1.

Options such as GROUPN= and TURNHLABELS are specified after the slash (/) in the PCHART statement. A complete list of options is presented in the "Syntax" section on page 63.

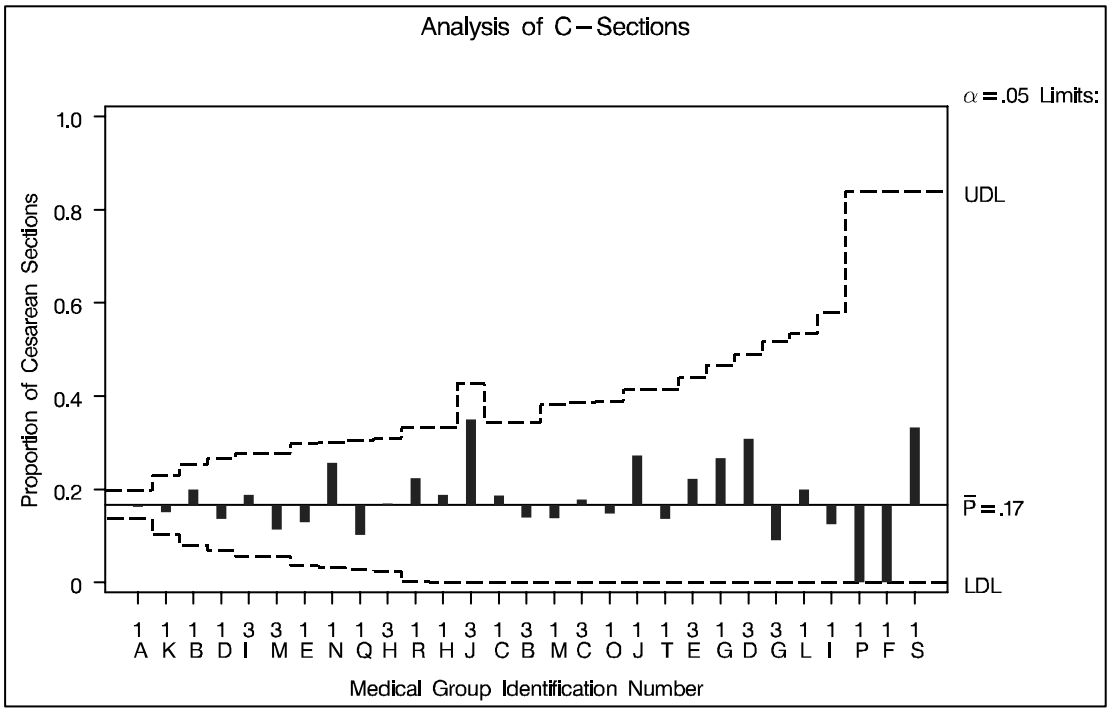


Figure 3.2. ANOM p Chart for Cesarean Sections

Each point on the p chart represents the proportion of c-sections for a particular group. For instance, the value plotted for group 1A is $150/923 = 0.163$.

Since all the points fall within the decision limits, it can be concluded that the variation in proportions of c-sections across medical groups is strictly due to chance.

By default, the decision limits shown correspond to a significance level of $\alpha = 0.05$. This means that, assuming all groups have the same proportion of c-sections, there is a 0.05 probability that one or more of the decision limits would be exceeded purely by chance. The formulas for the limits are given in "Decision Limits" on page 73.

Note that the decision limits vary with the number of deliveries in each group, and the widest limits correspond to the group with the smallest number of deliveries.

For more details on reading group counts, see "DATA= Data Set" on page 76.

Creating ANOM Charts for Proportions from Group Summary Data

See ANMPGRP in the SAS/QC Sample Library

The previous example illustrates how you can create ANOM charts for proportions using count data. However, in many applications, the group data are provided in summarized form as proportions or percentages. This example illustrates how you can use the PCHART statement with data of this type.

The following data set provides the data from the preceding example in summarized form:

```
data CsectProp;
   length ID $ 2;
   input ID CsectionsP CsectionsN @@;
datalines;
1A  0.163  923   1K  0.151  298   1B  0.200  170   1D  0.136  132
3I  0.189  106   3M  0.114  105   1E  0.130   77   1N  0.257   74
1Q  0.101   69   3H  0.169   65   1R  0.224   49   1H  0.188   48
3J  0.350   20   1C  0.186   43   3B  0.140   43   1M  0.138   29
3C  0.179   28   1O  0.148   27   1J  0.273   22   1T  0.136   22
3E  0.222   18   1G  0.267   15   3D  0.308   13   3G  0.091   11
1L  0.200   10   1I  0.125    8   1P  0.000    3   1F  0.000    3
1S  0.333    3
;
run;
```

A partial listing of CsectProp is shown in Figure 3.3. The groups are still indexed by ID. The variable CsectionsP contains the proportions of c-sections, and the variable CsectionsN contains the group sample sizes.

Proportions of Cesarean Sections

ID	Csections P	Csections N
1A	0.163	923
1K	0.151	298
1B	0.200	170
1D	0.136	132
3I	0.189	106
3M	0.114	105
1E	0.130	77
1N	0.257	74
1Q	0.101	69
3H	0.169	65

Figure 3.3. The Data Set CsectProp

You can analyze this data set by specifying it as a SUMMARY= data set in the PROC ANOM statement.

Note that Csections is *not* the name of a SAS variable in the data set but is, instead, the common prefix for the names of the two SAS variables CsectionsP and CsectionsN. The suffix characters *P* and *N* indicate *proportion* and *sample size*, respectively. Thus, you can specify two group variables in a SUMMARY= data set with a single name Csections, which is referred to as the *response*. The name ID specified after the asterisk is the name of the *group-variable*.

A SUMMARY= data set used with the PCHART statement must contain the following variables:

- group variable
- group proportion variable
- group sample size variable

Furthermore, the names of the group proportion and sample size variables must begin with the *response* name specified in the PCHART statement and end with the special suffix characters P and N, respectively.

For more information, see "SUMMARY= Data Set" on page 78.

The following statements create a p Chart for C-Sections using the SUMMARY= data set CsectProp:

```
title 'ANOM for the Proportion of Cesarean Sections';
proc anom summary=CsectProp;
   pchart Csections*ID / turnhlabels;
run;
```

The resulting ANOM p chart is shown in Figure 3.4. The TURNHLABELS option turns the horizontal axis labels since the default labeling skips labels if the characters exceed the space alloted. See Axis and Axis Label Options on page 66. To angle the horizontal axis labels, see Example 3.1.

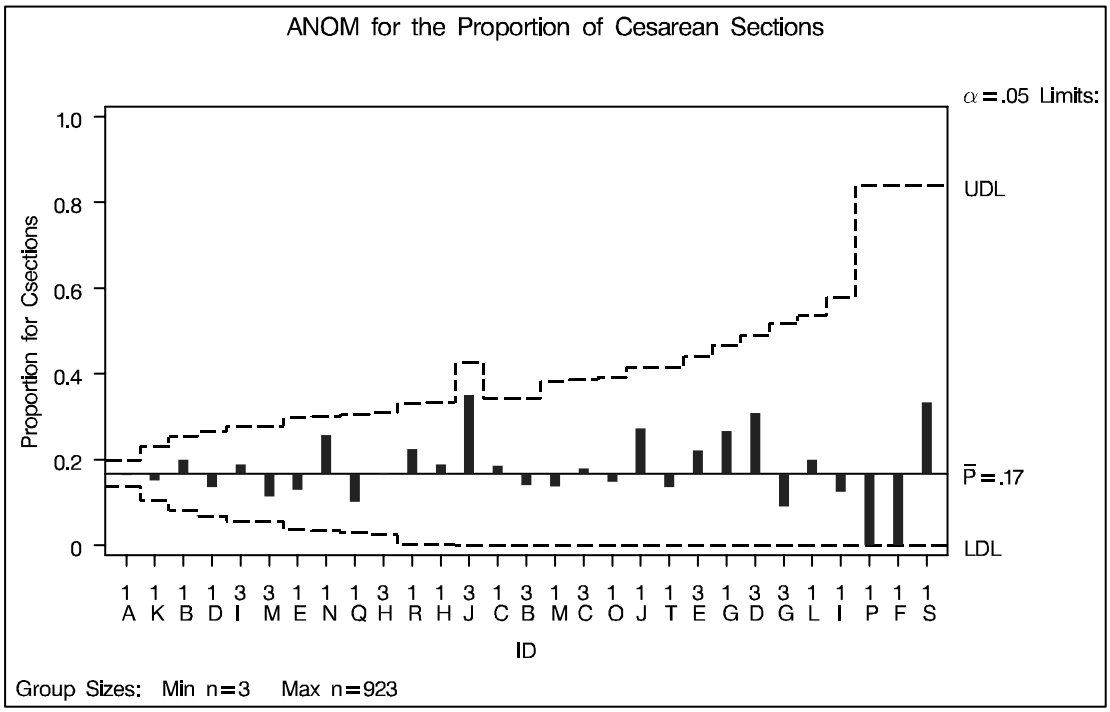


Figure 3.4. ANOM p Chart from Group Proportions

Saving Group Proportions

See ANMPSUM in the SAS/QC Sample Library

In this example, the PCHART statement is used to create a summary data set that can later be read by the ANOM procedure (as in the preceding example). The following statements read the data set CSection (see page 55) and create a summary data set named CSummary:

```
proc anom data=Csection;
   pchart Csections*ID / groupn     = Total
                         outsummary = CSummary
                         nochart ;
run;
```

The OUTSUMMARY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 3.2. Figure 3.5 contains a partial listing of CSummary.

Group Proportions and Decision Limit Information

ID	Csections P	Csections N
1A	0.16251	923
1K	0.15101	298
1B	0.20000	170
1D	0.13636	132
3I	0.18868	106
3M	0.11429	105
1E	0.12987	77
1N	0.25676	74
1Q	0.10145	69
3H	0.16923	65

Figure 3.5. The Data Set CSummary

There are three variables in the data set CSummary:

- ID identifies the groups.
- CSectionsP contains the group proportions.
- CSectionsN contains the group sample sizes.

Note that the variables containing the group proportions and group sample sizes are named by adding the suffix characters *P* and *N* to the *response* CSections specified in the PCHART statement. In other words, the variable naming convention for OUTSUMMARY= data sets is the same as that for SUMMARY= data sets. For more information, see "OUTSUMMARY= Data Set" on page 75.

Saving Decision Limits

You can save the decision limits for an ANOM *p* chart in a SAS data set.

See ANMPLIM in the SAS/QC Sample Library

The following statements read the number of c-sections per group from the data set CSection (see page 55) and save the decision limits displayed in Figure 3.2 in a data set named CSectionLim:

```
proc anom data=Csection;
   pchart Csections*ID / groupn    = Total
                         outlimits = CsectionLim
                         nochart;
run;
```

The OUTLIMITS= option names the data set containing the decision limits, and the NOCHART option suppresses the display of the chart. The data set CSectionLim is listed in Figure 3.6.

Decision Limits for the Proportion of Cesarean Sections

VAR	_GROUP_	_TYPE_	_LIMITN_	_ALPHA_	_LDLP_	_P_	_UDLP_	_LIMITK_
Csections	ID	ESTIMATE	V	0.05	V	0.16680	V	29

Figure 3.6. The Data Set CSectionLim with Decision Limits

The data set CSectionLim contains one observation with the limits for the *response* CSections. The variables _LDLP_ and _UDLP_ contain the lower and upper decision limits, and the variable _P_ contains the central line. The value of _LIMITN_ is the nominal sample size associated with the decision limits, the value of _LIMITK_ is the number of groups, and the value of _ALPHA_ is the significance level associated with the decision limits. The variables _VAR_ and _GROUP_ are bookkeeping variables that save the *response* and *group-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the value of _P_ is an estimate or a known (standard) value. Typically, the value of _TYPE_ is ESTIMATE.

For more information, see "OUTLIMITS= Data Set" on page 74.

See ANMPTAB in the SAS/QC Sample Library

You can create an output data set containing both decision limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc anom data=Csection;
   pchart Csections*ID / groupn   = Total
                         outtable = CsectionTab
                         nochart;
run;
```

A partial listing of the data set CSectionTab is shown in Figure 3.7.

Proportions and Decision Limits for Cesarean Sections

VAR	ID	_ALPHA_	_LIMITN_	_SUBN_	_LDLP_	_SUBP_	_P_	_UDLP_	_EXLIM_
Csections	1A	0.05	923	923	0.13658	0.16251	0.16680	0.19703	
Csections	1K	0.05	298	298	0.10356	0.15101	0.16680	0.23005	
Csections	1B	0.05	170	170	0.08060	0.20000	0.16680	0.25301	
Csections	1D	0.05	132	132	0.06816	0.13636	0.16680	0.26545	
Csections	3I	0.05	106	106	0.05610	0.18868	0.16680	0.27751	
Csections	3M	0.05	105	105	0.05555	0.11429	0.16680	0.27806	
Csections	1E	0.05	77	77	0.03611	0.12987	0.16680	0.29750	
Csections	1N	0.05	74	74	0.03340	0.25676	0.16680	0.30021	
Csections	1Q	0.05	69	69	0.02851	0.10145	0.16680	0.30510	
Csections	3H	0.05	65	65	0.02419	0.16923	0.16680	0.30941	

Figure 3.7. The Data Set CSectionTab

This data set contains one observation for each group sample. The variables _SUBP_ and _SUBN_ contain the group proportions and group sample sizes. The variables _LDLP_ and _UDLP_ contain the lower and upper decision limits, and

the variable _P_ contains the central line. The variables _VAR_ and ID contain the *response* name and values of the *group-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 76.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read the information in CSectionTab and display an ANOM p chart (not shown here) identical to the chart in Figure 3.2:

```
title 'Analysis of C-Sections';
proc anom table=CSectionTab;
   pchart CSections*id;
label _subp_ = 'Proportion of Cesarean Sections';
run;
```

Because the ANOM procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized ANOM charts. For more information, see "TABLE= Data Set" on page 79.

Syntax

The basic syntax for the PCHART statement is as follows:

PCHART *response***group-variable* **;**

The general form of this syntax is as follows:

PCHART (*responses***)***group-variable* <**(***block-variables* **)** >
< =*symbol-variable* | =*'character'* > < **/** *options* >**;**

You can use any number of PCHART statements in the ANOM procedure. The components of the PCHART statement are described as follows.

response
responses
identify one or more responses to be analyzed. The specification of *response* depends on the input data set specified in the PROC ANOM statement.

- If response counts are read from a DATA= data set, *response* must be the name of the variable containing the counts. For an example, see "Creating ANOM Charts for Proportions from Group Summary Data" on page 58.
- If response proportions are read from a SUMMARY= data set, *response* must be the common prefix of the summary variables in the SUMMARY= data set. For an example, see "Creating ANOM Charts for Proportions from Group Summary Data" on page 58.
- If response proportions and decision limits are read from a TABLE= data set, *response* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Decision Limits" on page 61.

A *response* is required. If you specify more than one response, enclose the list in parentheses. For example, the following statements request distinct ANOM p charts for the responses rejects and reworks:

```
proc anom data=measures;
   pchart (rejects reworks)*sample / groupn=100;
run;
```

Note that when data are read from a DATA= data set, the GROUPN= option, which specifies group sample sizes, is required.

group-variable
is the variable that identifies groups in the data. The *group-variable* is required. In the preceding PCHART statement, sample is the group variable.

block-variables
are optional variables that identify sets of consecutive groups on the chart. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker used to plot proportions. Distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOLn statements.

options
control the analysis, enhance the appearance of the chart, save results in data sets, and so on. The "Summary of Options" section, which follows, lists all options by function.

Summary of Options

The following tables list the PCHART statement options by function. Many of these options are identical to options in the SHEWHART procedure, which are described in detail beginning on page 1851 in Chapter 53, "Dictionary of Options."

Table 3.1. Tabulation Options

TABLE	creates a basic table of group sample sizes, group proportions, and decision limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, and TABLEOUTLIM
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLEOUTLIM	augments basic table with columns indicating decision limits exceeded

Note that specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.

Table 3.2. Reference Line Options

CHREF=*color*	specifies color for lines requested by the HREF= option
CVREF=*color*	specifies color for lines requested by the VREF= option
HREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis
HREFDATA= *SAS-data-set*	specifies position of reference lines perpendicular to horizontal
HREFLABELS= *'label1'...'labeln'*	specifies labels for HREF= lines
HREFLABPOS=*n*	specifies position of HREFLABELS= labels
LHREF=*linetype*	specifies line type for HREF= lines
LVREF=*linetype*	specifies line type for VREF= lines
NOBYREF	specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to vertical axis
VREFLABELS= *'label1'...'labeln'*	specifies labels for VREF= lines
VREFLABPOS=*n*	specifies position of VREFLABELS= labels

Table 3.3. Block Variable Legend Options

BLOCKLABELPOS= *keyword*	specifies position of label for *block-variable* legend
BLOCKLABTYPE= *n*|*keyword*	specifies text size of *block-variable* legend
BLOCKPOS=*n*	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=*color*	specifies color for filling background in *block-variable* legend
CBLOCKVAR=*variable*| (*variables*)	specifies one or more variables whose values are colors for filling background of *block-variable* legend

Table 3.4. Axis and Axis Label Options

CAXIS=*color*	specifies color for axis lines and tick marks
CFRAME=*color*|(*color-list*)	specifies fill colors for frame for plot area
CTEXT=*color*	specifies color for tick mark values and axis labels
HAXIS=*values*|AXIS*n*	specifies major tick mark values for horizontal axis
HEIGHT=*value*	specifies height of axis label and axis legend text
HMINOR=*n*	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=*value*	specifies length of offset at both ends of horizontal axis
INTSTART=*value*	specifies first major tick mark value for numeric horizontal axis
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent group values is to be labeled on horizontal axis
NOTRUNC	suppresses vertical axis truncation at zero applied by default
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=*n*	specifies thinning factor for tick mark labels on horizontal axis
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=*values*|AXIS*n*	specifies major tick mark values for vertical axis
VMINOR=*n*	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=*value*	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis
WAXIS=*n*	specifies width of axis lines
YSCALE=PERCENT	scales vertical axis in percent units (rather than proportions)

Table 3.5. Options for Specifying Decision Limits

ALPHA=*value*	specifies significance level
LIMITK=*n*	specifies number of group
LIMITN=*n*|VARYING	specifies either nominal sample size for fixed decision limits or varying limits
NOREADLIMITS	computes decision limits for each *response* from the data rather than from a LIMITS= data set
P=*value*	specifies the weighted average of group proportions
READINDEXES=ALL|'*label1*'...'*labeln*'	reads multiple sets of decision limits for each *response* from a LIMITS= data set
TYPE=*keyword*	identifies whether P= value is an estimate or standard value and specifies value of _TYPE_ in OUTLIMITS= data set

Table 3.6. Options for Displaying Decision Limits

CINFILL=*color*	specifies color for area inside decision limits
CLIMITS=*color*	specifies color of decision limits, central line, and related labels
LDLLABEL='*label*'	specifies label for lower decision limit
LIMLABSUBCHAR='*character*'	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the decision limit
LLIMITS=*linetype*	specifies line type for decision limits
NDECIMAL=*n*	specifies number of digits to right of decimal place in default labels for decision limits and central line
NOCTL	suppresses display of central line
NOLDL	suppresses display of lower decision limit
NOLIMITLABEL	suppresses labels for decision limits and central line
NOLIMITS	suppresses display of decision limits
NOLIMITSFRAME	suppresses default frame around decision limit information when multiple sets of decision limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for decision limits
NOLIMIT0	suppresses display of lower decision limit if it is 0
NOLIMIT1	suppresses display of upper decision limit if it is 1 (100%)
NOUDL	suppresses display of upper decision limit
PSYMBOL='*string*' *keyword*	specifies label for central line
UDLLABEL='*string*'	specifies label for upper decision limit
WLIMITS=*n*	specifies width for decision limits and central line

Table 3.7. Grid Options

GRID	adds grid to ANOM chart
LGRID=*linetype*	specifies line type for grid requested with the GRID option
WGRID=*n*	specifies width of grid lines

Table 3.8. Options for Plotting and Labeling Points

ALLLABEL=VALUE|(*variable*)	labels every point
CCONNECT=*color*	specifies color for line segments that connect points on chart
CFRAMELAB=*color*	specifies fill color for frame around labeled points
CNEEDLES=*color*	specifies color for needles that connect points to central line
COUT=*color*	specifies color for portions of line segments that connect points outside decision limits
COUTFILL=*color*	specifies color for shading areas between the connected points and decision limits outside the limits
NOCONNECT	suppresses line segments that connect points on chart
NONEEDLES	suppresses vertical needles connecting points to central line
OUTLABEL=VALUE|(*variable*)	labels points outside decision limits
SYMBOLLEGEND=NONE|*name*	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=*keyword*	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=*n*	specifies width of needles

Table 3.9. Options for Interactive ANOM Charts

HTML=(*variable*)	specifies a variable whose values are URLs to be associated with groups
HTML_LEGEND=(*variable*)	specifies a variable whose values are URLs to be associated with symbols in the symbol legend
WEBOUT=*SAS-data-set*	creates an OUTTABLE= data set with additional graphics coordinate data

Table 3.10. Clipping Options

CCLIP=*color*	specifies color for plot symbol for clipped points
CLIPFACTOR=*value*	determines extent to which extreme points are clipped
CLIPLEGEND='*string*'	specifies text for clipping legend
CLIPLEGPOS=*keyword*	specifies position of clipping legend
CLIPSUBCHAR='*character*'	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=*symbol*	specifies plot symbol for clipped points
CLIPSYMBOLHT=*value*	specifies symbol marker height for clipped points

Table 3.11. Phase Options

CPHASELEG=*color*	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE='*string*'	specifies value of _PHASE_ in the OUTSUMMARY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=*value*| *keyword*	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels decision limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES= ALL| '*label1*'...'*labeln*'	specifies *phases* to be read from an input data set

Table 3.12. Input Data Set Options

DATAUNIT=*keyword*	specifies that input values are proportions or percentages rather than counts
GROUPN=*n*| *variable*	specifies group sample sizes as constant number n or as values of *variable* in a DATA= data set

Table 3.13. Output Data Set Options

OUTSUMMARY= *SAS-data-set*	creates output data set containing group proportions and group sample sizes
OUTINDEX='*string*'	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS= *SAS-data-set*	creates output data set containing decision limits
OUTTABLE= *SAS-data-set*	creates output data set containing group proportions, group non-conforming items, group sample sizes, and decision limits

Table 3.14. Plot Layout Options

ALLN	plots proportions for all groups
BILEVEL	creates ANOM charts using half-screens and half-pages
EXCHART	creates ANOM charts for a *response* only when exceptions occur
MAXPANELS=*n*	specifies maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed decision limits
NOCHART	suppresses creation of chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for group sample sizes
NPANELPOS=*n*	specifies number of group positions per panel on each chart
REPEAT	repeats last group position on panel as first group position of next panel
TOTPANELS=*n*	specifies number of pages or screens to be used to display chart
ZEROSTD	displays p chart regardless of whether the average of the group proportions is zero

Table 3.15. Graphical Enhancement Options

ANNOTATE= *SAS-data-set*	specifies annotate data set that adds features to chart
DESCRIPTION='*string*'	specifies string that appears in the description field of the PROC GREPLAY master menu
FONT=*font*	specifies software font for labels and legends on charts
NAME='*string*'	specifies name that appears in the name field of the PROC GREPLAY master menu
PAGENUM='*string*'	specifies the form of the label used in pagination
PAGENUMPOS= *keyword*	specifies the position of the page number requested with the PAGENUM= option

Table 3.16. Star Options

CSTARCIRCLES=*color*	specifies color for circles specified by the STARCIRCLES= option
CSTARFILL=*color*| *(variable)*	specifies color for filling stars
CSTAROUT=*color*	specifies outline color for stars exceeding inner or outer circles
CSTARS=*color*| *(variable)*	specifies color for outlines of stars
LSTARCIRCLES= *linetypes*	specifies line types for STARCIRCLES= circles
LSTARS=*linetype*| *(variable)*	specifies line types for outlines of stars requested with the STARVERTICES= option
STARBDRADIUS=*value*	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=*value-list*	specifies reference circles for stars
STARINRADIUS=*value*	specifies inner radius of stars
STARLABEL=*keyword*	specifies vertices to be labeled
STARLEGEND=*keyword*	specifies style of legend for star vertices
STARLEGENDLAB= *'label'*	specifies label for STARLEGEND= legend
STAROUTRADIUS=*value*	specifies outer radius of stars
STARSPEC=*value*| *SAS-data-set*	specifies method used to standardize vertex variables
STARSTART=*value*	specifies angle for first vertex
STARTYPE=*keyword*	specifies graphical style of star
STARVERTICES= *variable*|*(variables)*	superimposes star at each point on chart
WSTARCIRCLES=*n*	specifies width of circles requested by the STARCIRCLES= option
WSTARS=*n*	specifies width of stars requested by the STARVERTICES= option

Table 3.17. Overlay Options

CCOVERLAY=*color-list*	specifies colors for overlay line segments
COVERLAY=*color-list*	specifies colors for overlay plots
COVERLAYCLIP=*color*	specifies color for clipped points on overlays
LOVERLAY=*linetypes*	specifies line types for overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=*variable-list*	specifies variables to overlay on ANOM chart
OVERLAYCLIPSYM=*symbol*	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=*value*	specifies symbol height for clipped points on overlays
OVERLAYHTML=*variable-list*	specifies URLS to associate with overlay points
OVERLAYLEGLAB='*label*'	specifies label for overlay legend
OVERLAYSYM=*symbol-list*	specifies symbols for overlay plots
OVERLAYSYMHT=*value-list*	specifies symbol heights for overlay plots
WOVERLAY=*value-list*	specifies widths of overlay line segments

Details

Constructing ANOM Charts for Proportions

The following notation is used in this section:

X_i	response number (count) in the ith group
k	number of groups
n_i	sample size of the ith group
N	total sample size $= n_1 + \cdots + n_k$
p_i	proportion in the ith group, where $p_i = X_i/n_i$
$\bar{p}$	weighted average of proportions across groups: $\bar{p} = \frac{n_1 p_1 + \cdots + n_k p_k}{N} = \frac{X_1 + \cdots + X_k}{N}$
α	significance level
$h(\alpha; k, n, \infty)$	critical value for ANOM for normal data in the balanced case ($n_i \equiv n$)
$h(\alpha; k, n_1, \ldots, n_k, \infty)$	critical value for ANOM for normal data in the unbalanced case

Plotted Points

Each point on an ANOM p chart represents the response proportion ($p_i = X_i/n_i$) for a group.

Central Line

By default, the central line on an ANOM p chart is computed as $\bar{p}$, the weighted average of the group proportions. You can specify $\bar{p}$ with the P= option or with the variable _P_ in a LIMITS= data set.

Decision Limits

For the ith group, the response counts are assumed to have the binomial distribution $B(n_i, p_i)$. The ANOM method for proportions tests the null hypothesis that $p_1 = p_2 = \cdots = p_k$, that is, that the proportions are the same, against the alternative that at least one of the p_i's is different.

The decision limits are computed using the normal approximation to the binomial distribution, which is appropriate when the sample sizes for the groups are large; refer to Ramig (1983). A commonly recommended check for this assumption is that $n_i p_i > 5$ and $n_i(1 - p_i) > 5$ for all the groups. The critical values in the ANOM method for normally distributed data are adapted to the binomial case by using infinite degrees of freedom for the variance.

When the sample sizes are constant across groups ($n_i \equiv n$), the decision limits are computed as follows:

$$\begin{aligned}
\text{lower decision limit (LDL)} &= \max\left(\bar{p} - h(\alpha; k, n, \infty)\sqrt{\bar{p}(1-\bar{p})}\sqrt{\frac{k-1}{nk}}\,, 0\right) \\
\text{upper decision limit (UDL)} &= \min\left(\bar{p} + h(\alpha; k, n, \infty)\sqrt{\bar{p}(1-\bar{p})}\sqrt{\frac{k-1}{nk}}\,, 1\right)
\end{aligned}$$

For the theoretical derivation of the decision limits, refer to Nelson (1982a).

When the sample sizes (n_i) are different across groups (the unbalanced case), the decision limits are computed as follows:

$$\begin{aligned}
\text{lower decision limit (LDL)} &= \max\left(\bar{p} - h(\alpha; k, n_1, \ldots, n_k, \infty)\sqrt{\bar{p}(1-\bar{p})}\sqrt{\frac{N-n_i}{Nn_i}}\,, 0\right) \\
\text{upper decision limit (UDL)} &= \min\left(\bar{p} + h(\alpha; k, n_1, \ldots, n_k, \infty)\sqrt{\bar{p}(1-\bar{p})}\sqrt{\frac{N-n_i}{Nn_i}}\,, 1\right)
\end{aligned}$$

Note that the decision limits for the ith group depend on n_i. If the sample sizes are constant across groups ($n_i \equiv n$), the decision limits in the unbalanced case reduce to the formulas given for the balanced case since $n_i = n$ and $N = kn$ so

$$\sqrt{\frac{N-n_i}{Nn_i}} = \sqrt{\frac{kn-n}{Nn}} = \sqrt{\frac{k-1}{N}} = \sqrt{\frac{k-1}{kn}}$$

For the derivation of the decision limits for unequal sample sizes, refer to Nelson (1991).

Exact critical values $h(\alpha; k, n, \infty)$ and $h(\alpha; k, n_1, \ldots, n_k, \infty)$ were first tabulated by L. S. Nelson (1983). Refer to Nelson (1993) for derivation of critical values.

You can specify parameters for the limits as follows:

- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set. By default, α = 0.05.
- Specify a constant nominal sample size $n_i \equiv n$ for the decision limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set. By default, n is the observed sample size in the balanced case.
- Specify $\bar{p}$ with the P= option or with the variable _P_ in a LIMITS= data set. By default, $\bar{p}$ is the weighted average of the group proportions.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves decision limits and decision limit parameters. The following variables can be saved:

Table 3.18. OUTLIMITS= Data Set

Variable	Description
ALPHA	significance level (α)
GROUP	*group-variable* specified in the PCHART statement
INDEX	optional identifier for the decision limits specified with the OUTINDEX= option
LDLP	lower decision limit for proportions
LIMITK	number of groups
LIMITN	nominal sample size associated with the decision limits
_P_	average proportion of nonconforming items ($\bar{p}$)
TYPE	type (standard or estimate) of _P_
UDLP	upper decision limit for proportions
VAR	*response* specified in the PCHART statement

Notes:

1. If the decision limits vary with group sample size, the special missing value V is assigned to the variables _LIMITN_, _LDLP_, and _UDLP_.

2. A group must have at least one nonmissing value ($n_i \geq 1$), and there must be at least one group with $n_i \geq 2$.
3. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *response* specified in the PCHART statement. For an example, see "Saving Decision Limits" on page 61.

OUTSUMMARY= Data Set

The OUTSUMMARY= data set saves group summary statistics. The following variables are saved:

- the *group-variable*
- a group proportion variable named by *response* suffixed with *P*
- a group sample size variable named by *response* suffixed with *N*

Given a *response* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Group summary variables are created for each *response* specified in the PCHART statement. For example, consider the following statements:

```
proc anom data=input;
   pchart (rework rejected)*batch / outsummary=summary
                                    groupn =30;
run;
```

The data set summary contains variables named batch, reworkP, reworkN, rejetedP, and rejetedN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTSUMMARY= data set, see "Saving Group Proportions" on page 60.

Note that an OUTSUMMARY= data set created with the PCHART statement can be reused as a SUMMARY= data set.

OUTTABLE= Data Set

The OUTTABLE= data set saves group summary statistics, decision limits, and related information. The variables shown in the following table are saved:

Variable	Description
ALPHA	significance level (α)
EXLIM	decision limit exceeded on p chart
group	values of the group variable
LDLP	lower decision limit for proportions
LIMITN	nominal sample size associated with the decision limits
SUBP	group proportion
SUBN	group sample size
UDLP	upper decision limit for proportions
VAR	*response* specified in the PCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. The variable _EXLIM_ is a character variable of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Decision Limits" on page 61.

ODS Tables

The following table summarizes the ODS tables that you can request with the PCHART statement.

Table 3.19. ODS Tables Produced with the PCHART Statement

Table Name	Description	Options
PCHART	p chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLEOUT

Input Data Sets

DATA= Data Set

You can read count data from a DATA= data set specified in the PROC ANOM statement. Each *response* specified in the PCHART statement must be a SAS variable in

the DATA= data set. This variable provides counts for group samples indexed by the values of the *group-variable*. The *group-variable*, which is specified in the PCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a count for each *response* and a value for the *group-variable*. The data set must contain one observation for each group. Note that you can specify the DATAUNIT= option in the PCHART statement to read proportions or percentages instead of counts. Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

When you use a DATA= data set with the PCHART statement, the GROUPN= option (which specifies the group sample size) is required.

For an example of a DATA= data set, see "Creating ANOM Charts for Proportions from Group Counts" on page 55.

LIMITS= Data Set

You can read preestablished decision limits (or parameters from which the decision limits can be calculated) from a LIMITS= data set specified in the PROC ANOM statement. For example, the following statements read decision limit information from the data set conlims:

```
proc anom data=info limits=conlims;
   pchart rejects*batch / groupn= 100;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the ANOM procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LDLP_, _P_, and _UDLP_, which specify the decision limits directly
- the variable _P_, without providing _LDLP_ and _UDLP_. The value of _P_ is used to calculate the decision limits according to the equations on page 73.

In addition, note the following:

- The variables _VAR_ and _GROUP_ are always required. These must be character variables whose lengths are no greater than 32.

- _LDLP_ and _UDLP_ must be specified together; otherwise their values are computed.
- _ALPHA_ is optional but is recommended in order to maintain a complete set of decision limit information. The default value is 0.05.
- _LIMITK_ is optional. The default value is k, the number of groups. A group must have at least one nonmissing value ($n_i \geq 1$) and there must be at least one group with $n_i \geq 2$. If specified, _LIMITK_ overrides the value of k.
- _LIMITN_ is optional. The default value is the common group size (n), in the balanced case $n_i \equiv n$. If specified, _LIMITN_ overrides the value of n.
- The variable _TYPE_ is optional, but is recommended to maintain a complete set of decision limit information. The variable _TYPE_ must be a character variable of length 8. Valid values are `ESTIMATE`, `STANDARD`, `STDMEAN`, and `STDRMS`. The default is `ESTIMATE`.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- BY variables are required if specified with a BY statement.

SUMMARY= Data Set

You can read group summary statistics from a SUMMARY= data set specified in the PROC ANOM statement. This allows you to reuse OUTSUMMARY= data sets that have been created in previous runs of the ANOM procedure or to create your own SUMMARY= data set.

A SUMMARY= data set used with the PCHART statement must contain the following:

- the *group-variable*
- a group proportion variable for each *response*
- a group sample size variable for each *response*

The names of the proportion sample size variables must be the *response* name concatenated with the special suffix characters P and N, respectively.

For example, consider the following statements:

```
proc anom summary=summary;
   pchart (rework rejected)*batch / groupn=50;
run;
```

The data set summary must include the variables batch, reworkP, reworkN, rejetedP, and rejetedN.

Note that if you specify a *response* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *response* name, suffixed with the appropriate character.

Other variables that can be read from a SUMMARY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

For an example of a SUMMARY= data set, see "Creating ANOM Charts for Proportions from Group Summary Data" on page 58.

TABLE= Data Set

You can read summary statistics and decision limits from a TABLE= data set specified in the PROC ANOM statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the ANOM procedure. Because the ANOM procedure simply displays the information read from a TABLE= data set, you can use TABLE= data sets to create specialized ANOM charts.

The following table lists the variables required in a TABLE= data set used with the PCHART statement:

Table 3.20. Variables Required in a TABLE= Data Set

Variable	Description
group-variable	values of the *group-variable*
LDLP	lower decision limit for proportions
LIMITN	nominal sample size associated with the decision limits
_P_	average proportion of nonconforming items
SUBN	group sample size
SUBP	group proportion of nonconforming items
UDLP	upper decision limit for proportions

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _VAR_. This variable is required if more than one *response* is specified or if the data set contains information for more than one *response*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Decision Limits" on page 61.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*group-variable*
Vertical	DATA=	*response*
Vertical	SUMMARY=	group proportion variable
Vertical	TABLE=	_SUBP_

For example, the following sets of statements specify the label *Proportion Nonconforming* for the vertical axis of the p chart:

```
proc anom data=circuits;
   pchart fail*batch / groupn=50;
   label fail = 'Proportion Nonconforming';
run;

proc anom summary=cirhist;
   pchart fail*batch ;
   label failp = 'Proportion Nonconforming';
run;

proc anom table=cirtable;
   pchart fail*batch ;
   label _SUBP_ = 'Proportion Nonconforming';
run;
```

In this example, the label assignments are in effect only for the duration of the procedure step, and they temporarily override any permanent labels associated with the variables.

Missing Values

An observation read from a DATA=, SUMMARY=, or TABLE= data set is not analyzed if the value of the group variable is missing. For a particular response variable, an observation read from a DATA= data set is not analyzed if the value of the response variable is missing. Missing values of response variables generally lead to unequal group sample sizes. For a particular response variable, an observation read from a SUMMARY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples

This section provides advanced examples of the PCHART statement.

Example 3.1. ANOM p Charts with Angled Axis Labels

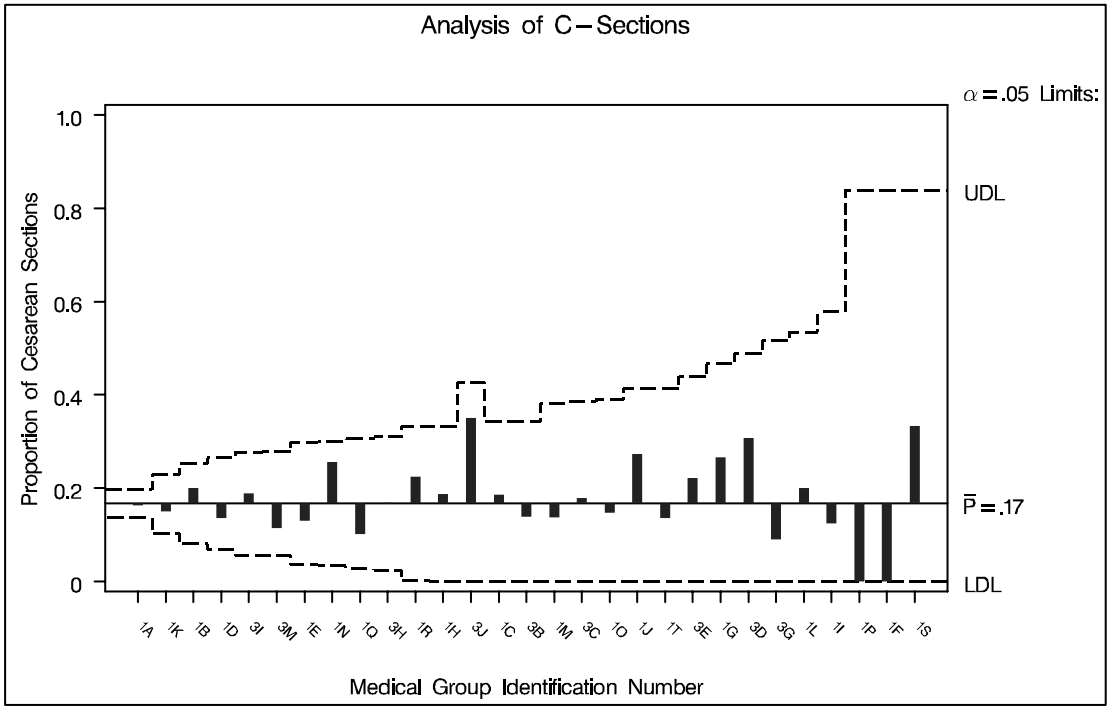
See ANMPEX1 in the SAS/QC Sample Library

Consider the example described in "Creating ANOM Charts for Proportions from Group Counts." In the example, the option TURNHLABELS was used to vertically display the horizontal axis labels. You can also use an AXIS statement to create an angled display of the horizontal or vertical axis labels. The following statements create the p CHART shown in Output 3.1.1:

```
title 'Analysis of C-Sections';
proc anom data=Csection;
   pchart Csections*ID / groupn   = Total
                         nolegend
                         haxis    = axis1;
   axis1 value     = (a=-45 h=2.0pct);
   label Csections = 'Proportion of Cesarean Sections';
run;
```

The angle is specified with the a= option in the AXIS1 statement. Valid angle values are between -90 and 90. The height of the labels is specified with the h= option in the AXIS1 statement. See Axis and Axis Label Options on page 66.

Output 3.1.1. ANOM p Chart for C-Sections with Angled Axis Labels



Chapter 4
UCHART Statement

Chapter Contents

OVERVIEW 85

GETTING STARTED 85
Creating ANOM Charts for Rates from Group Counts 86
Saving Decision Limits 88

SYNTAX 90
Summary of Options 91

DETAILS 99
Constructing ANOM Charts for Rates 99
Output Data Sets 101
ODS Tables 103
Input Data Sets 103
Axis Labels 106
Missing Values 106

EXAMPLES 107
Example 4.1. ANOM u Charts with Angled Axis Labels 107

Chapter 4
UCHART Statement

Overview

The UCHART statement creates ANOM charts for group (treatment level) rates, also referred to as ANOM *u charts*. The rate plotted on a *u* chart is the number or *count* of events occurring in a group divided by a measure of the opportunity for an event to occur.

You can use options in the UCHART statement to

- compute decision limits from the data based on specified parameters, such as the significance level (α)
- tabulate group summary statistics and decision limits
- save decision limits in an output data set
- save group summary statistics in an output data set
- read decision limits and decision limit parameters from a data set
- display distinct sets of decision limits for different sets of groups on the same chart
- add block legends and symbol markers to identify special groups
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

Getting Started

This section introduces the UCHART statement with simple examples that illustrate commonly used options. Complete syntax for the UCHART statement is presented in the "Syntax" section on page 90.

Creating ANOM Charts for Rates from Group Counts

See ANMU1 in the SAS/QC Sample Library

A health care system administrator uses ANOM to compare medical/surgical admissions rates for set of clinics. For more background concerning this application, refer to Rodriguez (1996).

The following statements create a SAS data set named MSAdmits, which contains the number of admissions and the number of member-months for each clinic during a one-year period.

```
data MSAdmits;
   length ID $ 2;
   input ID Count MemberMonths @@;
   KMemberYrs = MemberMonths/12000;
   label ID = 'Medical Group Id Number';
   datalines;
1A   1882    697204 1K    600    224715 1B    438    154720
1D    318     82254 3M    183     76450 3I    220     73529
1N    121     60169 3H    105     52886 1Q    124     52595
1E    171     51229 3B     88     34775 1C    100     31959
1H    112     28782 3C     84     27478 1R     69     26494
1T     21     25096 1M    130     24723 1O     61     24526
3D     66     22359 1J     54     19101 3J     30     16089
3G     36     13851 3E     26     10587 1G     28     10351
1I     25      6041 1L     20      5138 1S      7      2723
1F      7      2424 1P      2      2030
;
proc sort data=MSAdmits;
   by ID;
run;
```

A partial listing of MSAdmits is shown in Figure 4.1.

Medical/Surgical Admissions

ID	Count	Member Months	KMember Yrs
1A	1882	697204	58.1003
1B	438	154720	12.8933
1C	100	31959	2.6633
1D	318	82254	6.8545
1E	171	51229	4.2691
1F	7	2424	0.2020
1G	28	10351	0.8626
1H	112	28782	2.3985
1I	25	6041	0.5034
1J	54	19101	1.5918

Figure 4.1. The Data Set MSAdmits

There is a single observation per clinic. The variable ID identifies the clinics and is referred to as the *group-variable*. The variable Count provides the number of admissions for each clinic, which is referred to as the *response variable* (or *response*

for short). The variable MemberMonths, which provides the number of member-months for each clinic, is divided by 1200 to compute the variable KMemberYrs, the number of 1000-member-years, which serves as the measure of opportunity for an admission to occur.

The following example illustrates the basic form of the UCHART statement. After the keyword UCHART, you specify the *response* to analyze (in this case, Count), followed by an asterisk and the *group-variable* (ID).

The following statements create the u chart shown in Figure 4.2:

```
title 'Analysis of Medical/Surgical Admissions';
proc anom data=MSAdmits;
   uchart Count*ID / groupn = KMemberYrs
                     cneedles = black
                     turnhlabels
                     nolegend;
   label Count = 'Admits per 1000 Member Years';
run;
```

The TURNHLABELS option is used to vertically display the horizontal axis labels. The GROUPN= option specifies the number of "occurrence opportunity" units in each group and is required if the input data set is a DATA= data set. In this example, 1000 member years represent one unit of opportunity. The number of units per group can be thought of as the group "sample size." You can use the GROUPN= option to specify one of the following:

- a constant number of units, which applies to all the groups
- an input variable name, which provides the number of units for each group (KMemberYrs in this example)

Options such as GROUPN= are specified after the slash (/) in the UCHART statement. A complete list of options is presented in the "Syntax" section on page 90.

The input data set is specified with the DATA= option in the PROC ANOM statement.

Each point on the u chart represents the rate of occurrence, computed as the count divided by the number of opportunity units. The points are displayed in the sort order for the *group-variable* ID. The chart shows that Clinics 1D, 1H, and 1M have significantly higher admissions rates, and Clinics 1N, 1T, and 3H have significantly lower admissions rates.

By default, the decision limits correspond to a significance level of $\alpha = 0.05$. This means that, assuming all clinics have the same rate of admissions, there is a 0.05 probability that one or more of the decision limits would be exceeded purely by chance. The formulas for the limits are given in "Decision Limits" on page 100. Note that the decision limits vary with the number of 1000-member-years for each clinic.

For more details on reading count data, see "DATA= Data Set" on page 103.

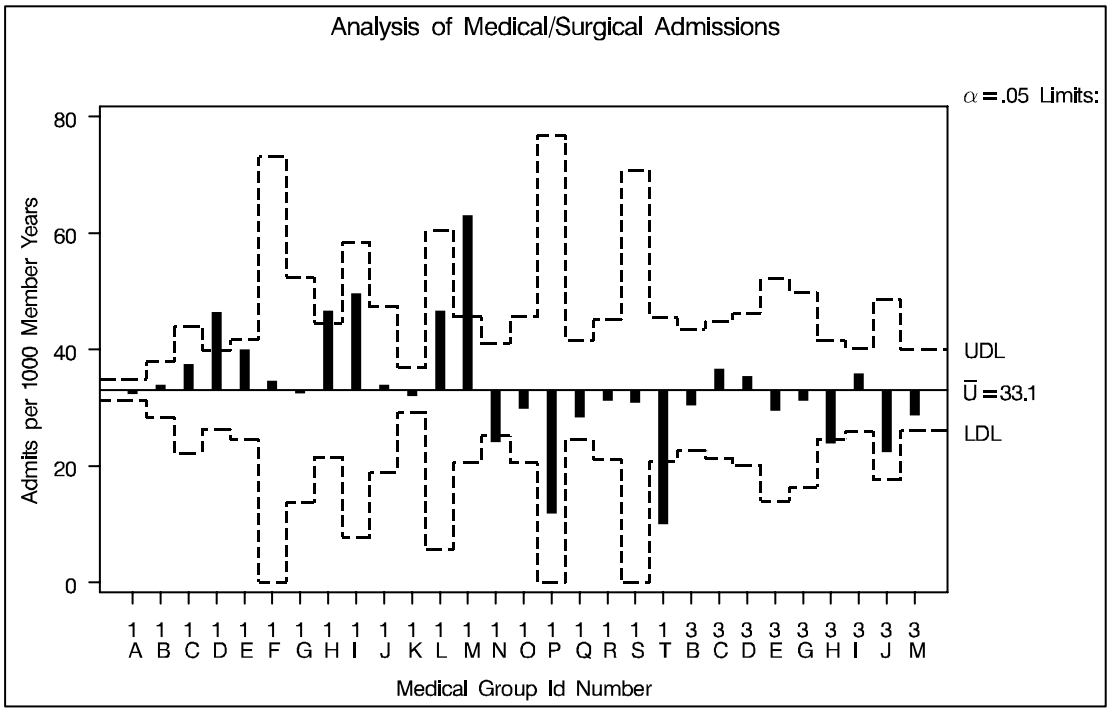


Figure 4.2. u Chart Example

Saving Decision Limits

See ANMULIM in the SAS/QC Sample Library

You can save the decision limits for an ANOM u chart in a SAS data set.

The following statements read the data set MSAdmits (see page 86) and save the decision limits displayed in Figure 4.2 in a data set named MSLimits:

```
proc anom data=MSAdmits;
   uchart Count*ID / groupn    = KMemberYrs
                     outlimits = MSLimits
                     nochart;
run;
```

The GROUPN= option specifies the number of opportunity units for each group. The OUTLIMITS= option names the data set containing the decision limits, and the NOCHART option suppresses the display of the chart. The data set MSLimits is listed in Figure 4.3.

Decision Limits for Medical/Surgical Admissions Rates

VAR	_GROUP_	_TYPE_	_LIMITN_	_ALPHA_	_LDLU_	_U_	_UDLU_	_LIMITK_
Count	ID	ESTIMATE	V	0.05	V	33.0789	V	29

Figure 4.3. Data Set MSLimits Containing Decision Limits

The data set MSLimits contains one observation with the limits for *response* Count. The variables _LDLU_ and _UDLU_ contain the lower and upper decision limits, and the variable _U_ contains the central line. The value of _LIMITN_ is the nominal number of units associated with the decision limits (which are varying in this case), the value of _LIMITK_ is the number of groups, and the value of _ALPHA_ is the significance level of the decision limits. The variables _VAR_ and _GROUP_ are bookkeeping variables that save the *response* and *group-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the value of _U_ is an estimate or standard (known) value. Typically, the value of _TYPE_ is ESTIMATE. For more information, see "OUTLIMITS= Data Set" on page 101.

Alternatively, you can use the OUTTABLE= option to create an output data set that saves both the decision limits and the group statistics, as illustrated by the following statements:

```
proc anom data=MSAdmits;
   uchart Count*ID / groupn   = KMemberYrs
                     outtable = MSTable
                     nochart;
run;
```

The a partial listing of the data set MSTable is shown in Figure 4.4.

Rates and Decision Limits for Medical/Surgical Admissions

VAR	ID	_ALPHA_	_LIMITN_	_SUBN_	_LDLU_	_SUBU_	_U_	_UDLU_	_EXLIM_
Count	1A	0.05	58.1003	58.1003	31.2138	32.3922	33.0789	34.9441	
Count	1B	0.05	12.8933	12.8933	28.2844	33.9710	33.0789	37.8735	
Count	1C	0.05	2.6633	2.6633	22.1566	37.5481	33.0789	44.0013	
Count	1D	0.05	6.8545	6.8545	26.3650	46.3929	33.0789	39.7928	UPPER
Count	1E	0.05	4.2691	4.2691	24.4976	40.0554	33.0789	41.6602	
Count	1F	0.05	0.2020	0.2020	0.0000	34.6535	33.0789	73.0573	
Count	1G	0.05	0.8626	0.8626	13.7738	32.4606	33.0789	52.3840	
Count	1H	0.05	2.3985	2.3985	21.5596	46.6959	33.0789	44.5983	UPPER
Count	1I	0.05	0.5034	0.5034	7.7793	49.6607	33.0789	58.3786	
Count	1J	0.05	1.5918	1.5918	18.9012	33.9249	33.0789	47.2566	

Figure 4.4. Data Set MSTable

This data set contains one observation for each group. The variables _SUBU_ and _SUBN_ contain the rate of occurrence and the number of opportunity units for each group. The variables _LDLU_ and _UDLU_ contain the lower and upper decision limits, and the variable _U_ contains the central line. The variables _VAR_ and ID contain the *response* name and values of the *group-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 102.

An OUTTABLE= data set can be read later as a TABLE= data set by the ANOM procedure. For example, the following statements read MSTable and display a u chart (not shown here) identical to the chart in Figure 4.2:

See ANMUTAB in the SAS/QC Sample Library

```
title 'Analysis of Medical/Surgical Admissions';
proc anom table=MSTable;
   uchart Count*id ;
   label _subu_ = 'Admits per 1000 Member Years';
run;
```

Because the ANOM procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized ANOM charts. For more information, see "TABLE= Data Set" on page 105.

Syntax

The basic syntax for the UCHART statement is as follows:

UCHART *response*******group-variable* **;**

The general form of this syntax is as follows:

UCHART (*responses***)****group-variable* <**(***block-variables* **)** > < =*symbol-variable* > < **/** *options* >**;**

You can use any number of UCHART statements in the ANOM procedure. The components of the UCHART statement are described as follows.

response
responses
identify one or more responses to be analyzed. The specification of *response* depends on the input data set specified in the PROC ANOM statement.

- If counts are read from a DATA= data set, *response* must be the name of the variable containing the counts. For an example, see "Creating ANOM Charts for Rates from Group Counts" on page 86.
- If rates and numbers of opportunity units are read from a SUMMARY= data set, *response* must be the common prefix of the appropriate variables in the SUMMARY= data set.
- If rates, numbers of opportunity units, and decision limits are read from a TABLE= data set, *response* must be the value of the variable _VAR_ in the TABLE= data set.

A *response* is required. If you specify more than one response, enclose the list in parentheses. For example, the following statements request distinct ANOM u charts for defects and flaws:

```
proc anom data=measures;
   uchart (defects flaws)*sample / groupn=30;
run;
```

Note that when data are read from a DATA= data set with the UCHART statement, the GROUPN= option (which specifies the number of opportunity units per group) is required.

group-variable
is the variable that identifies groups in the data. The *group-variable* is required. In the preceding UCHART statement, sample is the group variable.

block-variables
are optional variables that identify sets of consecutive groups on the chart. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker used to plot the rates. Distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOLn statements.

options
enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. "Summary of Options," which follows, lists all options by function.

Summary of Options

The following tables list the UCHART statement options by function. Many of these options are identical to options in the SHEWHART procedure, which are described in detail beginning on page 1851 in Chapter 53, "Dictionary of Options."

Table 4.1. Tabulation Options

TABLE	creates a basic table of rates, numbers of opportunity units, and decision limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, and TABLEOUTLIM
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLEOUTLIM	augments basic table with columns indicating decision limits exceeded

Note that specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.

Table 4.2. Reference Line Options

CHREF=*color*	specifies color for lines requested by the HREF= option
CVREF=*color*	specifies color for lines requested by the VREF= option
HREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on u chart
HREFDATA= *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on u chart
HREFLABELS= *'label1'...'labeln'*	specifies labels for HREF= lines
HREFLABPOS=*n*	specifies position of HREFLABELS= labels
LHREF=*linetype*	specifies line type for HREF= lines
LVREF=*linetype*	specifies line type for VREF= lines
NOBYREF	specifies that reference line information in a data set is to be applied uniformly to charts created for all BY groups
VREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to vertical axis
VREFLABELS= *'label1'...'labeln'*	specifies labels for VREF= lines
VREFLABPOS=*n*	specifies position of VREFLABELS= labels

Table 4.3. Clipping Options

CCLIP=*color*	specifies color for plot symbol for clipped points
CLIPFACTOR=*value*	determines extent to which extreme points are clipped
CLIPLEGEND='*string*'	specifies text for clipping legend
CLIPLEGPOS=*keyword*	specifies position of clipping legend
CLIPSUBCHAR= '*character*'	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=*symbol*	specifies plot symbol for clipped points
CLIPSYMBOLHT=*value*	specifies symbol marker height for clipped points

Table 4.4. Axis and Axis Label Options

CAXIS=*color*	specifies color for axis lines and tick marks
CFRAME=*color*|(*color-list*)	specifies fill colors for frame for plot area
CTEXT=*color*	specifies color for tick mark values and axis labels
HAXIS=*values*|AXIS*n*	specifies major tick mark values for horizontal axis
HEIGHT=*value*	specifies height of axis label and axis legend text
HMINOR=*n*	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=*value*	specifies length of offset at both ends of horizontal axis
NOHLABEL	suppresses label for horizontal axis
NOTRUNC	suppresses vertical axis truncation at zero applied by default
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=*n*	specifies thinning factor for tick mark labels on horizontal axis
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=*values*|AXIS*n*	specifies major tick mark values for vertical axis
VMINOR=*n*	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=*value*	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis for primary chart
WAXIS=*n*	specifies width of axis lines

Table 4.5. Options for Specifying Decision Limits

ALPHA=*value*	specifies significance level
LIMITN=*n*|VARYING	specifies either a nominal number of opportunity units for fixed decision limits or varying limits
NOREADLIMITS	computes decision limits for each *response* from the data rather than from a LIMITS= data set
READINDEXES=ALL| '*label1*'...'*labeln*'	reads multiple sets of decision limits for each *response* from a LIMITS= data set
TYPE=*keyword*	identifies whether U= value is an estimate or standard value and specifies the value of _TYPE_ in OUTLIMITS= data set
U=*value*	specifies the weighted average of group rates

Table 4.6. Options for Displaying Decision Limits

CINFILL=*color*	specifies color for area inside decision limits
CLIMITS=*color*	specifies color of decision limits, central line, and related labels
LDLLABEL='*label*'	specifies label for lower decision limit
LIMLABSUBCHAR= '*character*'	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the decision limit
LLIMITS=*linetype*	specifies line type for decision limits
NDECIMAL=*n*	specifies number of digits to right of decimal place in default labels for decision limits and central line
NOCTL	suppresses display of central line on u chart
NOLDL	suppresses display of lower decision limit
NOLIMITLABEL	suppresses labels for decision limits and central line
NOLIMITS	suppresses display of decision limits
NOLIMITSFRAME	suppresses default frame around decision limit information when multiple sets of decision limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for decision limits
NOLIMIT0	suppresses display of zero lower decision limit for u chart
NOUDL	suppresses display of upper decision limit
UDLLABEL='*string*'	specifies label for upper decision limit
USYMBOL='*string*'| *keyword*	specifies label for central line
WLIMITS=*n*	specifies width for decision limits and central line

Table 4.7. Grid Options

GRID	adds grid to ANOM chart
LGRID=*linetype*	specifies line type for grid requested with the GRID option
WGRID=*n*	specifies width of grid lines

Table 4.8. Options for Plotting and Labeling Points

ALLLABEL=VALUE|(*variable*)	labels every point
CCONNECT=*color*	specifies color for line segments that connect points on chart
CFRAMELAB=*color*	specifies fill color for frame around labeled points
CNEEDLES=*color*	specifies color for needles that connect points to central line
COUT=*color*	specifies color for portions of line segments that connect points outside decision limits
COUTFILL=*color*	specifies color for shading areas between the connected points and decision limits outside the limits
NOCONNECT	suppresses line segments that connect points on chart
NONEEDLES	suppresses vertical needles connecting points to central line
OUTLABEL=VALUE|(*variable*)	labels points outside decision limits
SYMBOLLEGEND=NONE|*name*	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=*keyword*	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=*n*	specifies width of needles

Table 4.9. Options for Interactive ANOM Charts

HTML=(*variable*)	specifies a variable whose values are URLs to be associated with groups
HTML_LEGEND=(*variable*)	specifies a variable whose values are URLs to be associated with symbols in the symbol legend
WEBOUT=*SAS-data-set*	creates an OUTTABLE= data set with additional graphics co-ordinate data

Table 4.10. Block Variable Legend Options

BLOCKLABELPOS=*keyword*	specifies position of label for *block-variable* legend
BLOCKLABTYPE=*value*|*keyword*	specifies text size of *block-variable* legend
BLOCKPOS=*n*	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=*color*	specifies color for filling background in *block-variable* legend
CBLOCKVAR=*variable*|(*variables*)	specifies one or more variables whose values are colors for filling background of *block-variable* legend

Table 4.11. Phase Options

CPHASELEG=*color*	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE='*string*'	specifies value of the variable _PHASE_ in the OUTSUMMARY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=*value*| *keyword*	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels decision limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES= ALL| '*label1*'...'*labeln*'	specifies *phases* to be read from an input data set

Table 4.12. Input Data Set Option

GROUPN=*n*| *variable*	specifies the number of opportunity units for groups as a constant number n or as values of *variable* in the DATA= data set

Table 4.13. Output Data Set Options

OUTSUMMARY= *SAS-data-set*	creates output data set containing rates and numbers of opportunity units for groups
OUTINDEX='*string*'	specifies value of the variable _INDEX_ in the OUTLIMITS= data set
OUTLIMITS= *SAS-data-set*	creates output data set containing decision limits
OUTTABLE= *SAS-data-set*	creates output data set containing rates, numbers of opportunity units, and decision limits for group

Table 4.14. Plot Layout Options

ALLN	plots numbers of nonconformities per unit for all groups
BILEVEL	creates ANOM charts using half-screens and half-pages
EXCHART	creates ANOM charts for a response variable only when exceptions occur
MAXPANELS=*n*	specifies maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed decision limits
NOCHART	suppresses creation of chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for group sizes
NPANELPOS=*n*	specifies number of group positions per panel on each chart
REPEAT	repeats last group position on panel as first group position of next panel
TOTPANELS=*n*	specifies number of pages or screens to be used to display chart
ZEROSTD	displays chart regardless of whether the average of the group rates is zero

Table 4.15. Graphical Enhancement Options

ANNOTATE= *SAS-data-set*	specifies annotate data set that adds features to chart
DESCRIPTION='*string*'	specifies string that appears in the description field of the PROC GREPLAY master menu
FONT=*font*	specifies software font for labels and legends on charts
NAME='*string*'	specifies name that appears in the name field of the PROC GREPLAY master menu
PAGENUM='*string*'	specifies the form of the label used in pagination
PAGENUMPOS= *keyword*	specifies the position of the page number requested with the PAGENUM= option

Table 4.16. Star Options

CSTARCIRCLES=*color*	specifies color for circles specified by the STARCIRCLES= option
CSTARFILL=*color*| *(variable)*	specifies color for filling stars
CSTAROUT=*color*	specifies outline color for stars exceeding inner or outer circles
CSTARS=*color*| *(variable)*	specifies color for outlines of stars
LSTARCIRCLES= *linetypes*	specifies line types STARCIRCLES= circles
LSTARS=*linetype*| *(variable)*	specifies line types for outlines of stars requested with the STARVERTICES= option
STARBDRADIUS=*value*	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=*value-list*	specifies reference circles for stars
STARINRADIUS=*value*	specifies inner radius of stars
STARLABEL=*keyword*	specifies vertices to be labeled
STARLEGEND=*keyword*	specifies style of legend for star vertices
STARLEGENDLAB= *'label'*	specifies label for STARLEGEND= legend
STAROUTRADIUS=*value*	specifies outer radius of stars
STARSPEC=*value*| *SAS-data-set*	specifies method used to standardize vertex variables
STARSTART=*value*	specifies angle for first vertex
STARTYPE=*keyword*	specifies graphical style of star
STARVERTICES= *variable*|*(variables)*	superimposes star at each point on chart
WSTARCIRCLES=*n*	specifies width of circles requested by the STARCIRCLES= option
WSTARS=*n*	specifies width of stars requested by the STARVERTICES= option

Table 4.17. Overlay Options

CCOVERLAY=*color-list*	specifies colors for overlay line segments
COVERLAY=*color-list*	specifies colors for overlay plots
COVERLAYCLIP=*color*	specifies color for clipped points on overlays
LOVERLAY=*linetypes*	specifies line types for overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=*variable-list*	specifies variables to overlay on ANOM chart
OVERLAYCLIPSYM=*symbol*	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=*value*	specifies symbol height for clipped points on overlays
OVERLAYHTML=*variable-list*	specifies URLS to associate with overlay points
OVERLAYLEGLAB='*label*'	specifies label for overlay legend
OVERLAYSYM=*symbol-list*	specifies symbols for overlay plots
OVERLAYSYMHT=*value-list*	specifies symbol heights for overlay plots
WOVERLAY=*value-list*	specifies widths of overlay line segments

Details

Constructing ANOM Charts for Rates

The following notation is used in this section:

c_i	count (number of occurrences) in the ith group
k	number of groups
n_i	number of occurrence opportunity units in the ith group
N	total sample size $= n_1 + \cdots + n_k$
u_i	occurrence rate in the ith group ($u_i = c_i/n_i$)
$\bar{u}$	average of occurrence rates taken across groups. The quantity $\bar{u}$ is computed as a weighted average: $$\bar{u} = \frac{n_1 u_1 + \cdots + n_k u_k}{N} = \frac{c_1 + \cdots + c_k}{N}$$
α	significance level
$h(\alpha; k, n, \infty)$	critical value for ANOM for normal data in the balanced case ($n_i \equiv n$)
$h(\alpha; k, n_1, \ldots, n_k, \infty)$	critical value for ANOM for normal data in the unbalanced case

Plotted Points

Each point on a u chart indicates the rate of occurrence (u_i) in a group.

Central Line

In an ANOM chart for rates, the central line represents the weighted average of the group rates, which is denoted by $\bar{u}$.

Decision Limits

For the ith group, the occurrence counts are assumed to have a Poisson distribution with parameter λ_i. The ANOM method tests the null hypothesis that $\lambda_1 = \cdots = \lambda_k$, that is, that the rates are the same, against the alternative that at least one of the λ_i's is different.

The decision limits are computed using the normal approximation to the Poisson distribution, which is appropriate when the sample sizes for the groups are large; see Ramig (1983). A commonly recommended check for this assumption is that $c_i > 5$ for all the groups. The critical values in the ANOM method for normally distributed data are adapted to the Poisson case by using infinite degrees of freedom for the variance.

When the number of opportunity units is constant ($n_i \equiv n$) across groups, the decision limits are computed as follow:

$$\begin{aligned} \text{lower decision limit (LDLU)} &= \max\left(\bar{u} - h(\alpha; k, n, \infty)\sqrt{\bar{u}}\sqrt{\frac{k-1}{nk}}\,, 0\right) \\ \text{upper decision limit (UDLU)} &= \bar{u} + h(\alpha; k, n, \infty)\sqrt{\bar{u}}\sqrt{\frac{k-1}{nk}} \end{aligned}$$

For the theoretical derivation of the decision limits, refer to Nelson (1982a).

When the number of opportunity units (n_i) is different across groups (the unbalanced case), the decision limits are computed as follows:

$$\begin{aligned} \text{lower decision limit (LDLU)} &= \max\left(\bar{u} - h(\alpha; k, n_1, \ldots, n_k, \infty)\sqrt{\bar{u}}\sqrt{\frac{N-n_i}{Nn_i}}\,, 0\right) \\ \text{upper decision limit (UDLU)} &= \bar{u} + h(\alpha; k, n_1, \ldots, n_k, \infty)\sqrt{\bar{u}}\sqrt{\frac{N-n_i 1}{Nn_i}} \end{aligned}$$

Note that the decision limits for the ith group depend on n_i. If the sample sizes are constant across groups ($n_i \equiv n$), the decision limits in the unbalanced case reduce to the formulas given for the balanced case, since $n_i \equiv n$ and $N = kn$, so

$$\sqrt{\frac{N-n_i}{Nn_i}} = \sqrt{\frac{kn-n}{Nn}} = \sqrt{\frac{k-1}{N}} = \sqrt{\frac{k-1}{kn}}$$

For the derivation of the decision limits for unequal sample sizes, refer to Nelson (1991).

Exact critical values were first tabulated by Nelson (1982a). Refer to Nelson (1993) for a derivation of the critical values $h(\alpha; k, n, \infty)$ and Nelson (1991) for a derivation of the critical values $h(\alpha; k, n_1, \ldots, n_k, \infty)$. Note that the critical values in the unequal sample size case have not been tabulated.

You can specify parameters for the limits as follows:

- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a nominal constant number of opportunity units $n_i \equiv n$ with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify $\bar{u}$ with the U= option or with the variable _U_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves decision limits and decision limit parameters. The following variables can be saved:

Table 4.18. OUTLIMITS= Data Set

Variable	Description
ALPHA	significance level (α)
GROUP	*group-variable* specified in the UCHART statement
INDEX	optional identifier for the decision limits specified with the OUTINDEX= option
LDLU	lower decision limit for occurrence rates
LIMITK	number of groups
LIMITN	number of opportunity units associated with the decision limits
TYPE	type (estimate or standard value) of _U_
_U_	value of central line of u chart ($\bar{u}$)
UDLU	upper decision limit for occurrence rates
VAR	*response* specified in the UCHART statement

Notes:

1. If the decision limits vary with the number of opportunity units, the special missing value *V* is assigned to the variables _LDLU_, _UDLU_, and _LIMITN_.
2. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *response* specified in the UCHART statement. For an example, see "Saving Decision Limits" on page 88.

OUTSUMMARY= Data Set

The OUTSUMMARY= data set saves group summary statistics. The following variables are saved:

- the *group-variable*
- a response rate variable, whose name is *response* suffixed with *U*
- a number of opportunity units variable, whose name is *response* suffixed with *N*

Given a *response* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Group summary variables are created for each *response* specified in the UCHART statement. For example, consider the following statements:

```
proc anom data=fabric;
   uchart (flaws ndefects)*treatment / outsummary=summary
                                       groupn = 30;
run;
```

The data set summary contains the variables treatment, flawsU, flawsN, ndefctsU, and ndefctsN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

OUTTABLE= Data Set

The OUTTABLE= data set saves group summary statistics, decision limits, and related information. The following variables are saved:

Variable	Description
ALPHA	significance level (α)
EXLIM	decision limit exceeded on u chart
group	values of the group variable
LDLU	lower decision limit for group rate
LIMITN	nominal number of opportunity units associated with the decision limits
SUBU	group rate
SUBN	number of opportunity units in group
UDLU	upper decision limit for group rate
VAR	*response* specified in the UCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Note: The variable _EXLIM_ is a character variable of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

ODS Tables

The following table summarizes the ODS tables that you can request with the UCHART statement.

Table 4.19. ODS Tables Produced with the UCHART Statement

Table Name	Description	Options
UCHART	ANOM u chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLEOUT

Input Data Sets

DATA= Data Set

You can read response counts for groups from a DATA= data set specified in the PROC ANOM statement. Each *response* specified in the UCHART statement must be a SAS variable in the data set. This variable provides the count (number of occurrences) for groups indexed by the *group-variable*. The *group-variable*, specified in the UCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *response* and a value for the *group-variable*. The data set should contain one observation per group. When you use a DATA= data set with the UCHART statement, the GROUPN= option (which specifies the number of inspection units per group) is required. Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

For an example of a DATA= data set, see "Creating ANOM Charts for Rates from Group Counts" on page 86.

LIMITS= Data Set

You can read decision limits (or parameters from which the decision limits can be calculated) from a LIMITS= data set specified in the PROC ANOM statement. For example, the following statements read decision limit information from the data set conlims:

```
proc anom data=info limits=conlims;
   uchart defects*treatment / groupn = 30;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the ANOM procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LDLU_, _U_, and _UDLU_, which specify the decision limits
- the variable _U_, without providing the variables _LDLU_ and _UDLU_, which is used to calculate the decision limits (see page 100)

In addition, note the following:

- The variables _VAR_ and _GROUP_ are always required. These must be character variables whose lengths are no greater than 32.
- _LDLU_ and _UDLU_ must be specified together; otherwise their values are computed.
- _ALPHA_ is optional but is recommended in order to maintain a complete set of decision limit information. The default value is 0.05.
- _LIMITK_ is optional. The default value is k, the number of groups. A group must have at least one nonmissing value ($n_i \geq 1$) and there must be at least one group with $n_i \geq 2$. If specified, _LIMITK_ overrides the value of k.
- _LIMITN_ is optional. The default value is the common group size (n), in the balanced case $n_i \equiv n$. If specified, _LIMITN_ overrides the value of n.
- The variable _TYPE_ is optional, but is recommended to maintain a complete set of decision limit information. The variable _TYPE_ must be a character variable of length 8. Valid values are `ESTIMATE`, `STANDARD`, `STDMEAN`, and `STDRMS`. The default is `ESTIMATE`.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- BY variables are required if specified with a BY statement.

SUMMARY= Data Set

You can read group summary statistics from a SUMMARY= data set specified in the PROC ANOM statement. This enables you to reuse OUTSUMMARY= data sets that have been created in previous runs of the ANOM procedure or to read output data sets created with SAS summarization procedures.

A SUMMARY= data set used with the UCHART statement must contain the following variables:

- *group-variable*
- response rates for each *response*
- number of occurrence units for each *response*

The names of the variables containing the rates and number of occurrence units must be the *response* name concatenated with the special suffix characters U and N, respectively. For example, consider the following statements:

```
proc anom summary=summary;
   uchart (flaws ndefects)*treatment;
run;
```

The data set summary must include the variables treatment, flawsU, flawsN, ndefctsU, and ndefctsN.

Note that if you specify a *response* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *response* name, suffixed with the appropriate character.

Other variables that can be read from a SUMMARY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

TABLE= Data Set

You can read group statistics and decision limits from a TABLE= data set specified in the PROC ANOM statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the ANOM procedure or to create your own TABLE= data set. Because the ANOM procedure simply displays the information read from a TABLE= data set, you can use TABLE= data sets to create specialized ANOM charts.

The following table lists the variables required in a TABLE= data set used with the UCHART statement:

Table 4.20. Variables Required in a TABLE= Data Set

Variable	Description
group-variable	values of the *group-variable*
LDLU	lower decision limit for rate
LIMITN	nominal number of opportunity units associated with the decision limits
SUBN	number of opportunity units in group
SUBU	response rate for group
_U_	weighted average of group rates
UDLU	upper decision limit for rate

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _VAR_. This variable is required if more than one *response* is specified or if the data set contains information for more than one *response*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Decision Limits" on page 88.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*group-variable*
Vertical	DATA=	*response*
Vertical	SUMMARY=	group defects per unit variable
Vertical	TABLE=	_SUBU_

Missing Values

An observation read from a DATA=, SUMMARY=, or TABLE= data set is not analyzed if the value of the group variable is missing. For a particular response variable, an observation read from a DATA= data set is not analyzed if the value of the response variable is missing. For a particular response variable, an observation read from a SUMMARY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples

This section provides an advanced example of the UCHART statement.

Example 4.1. ANOM u Charts with Angled Axis Labels

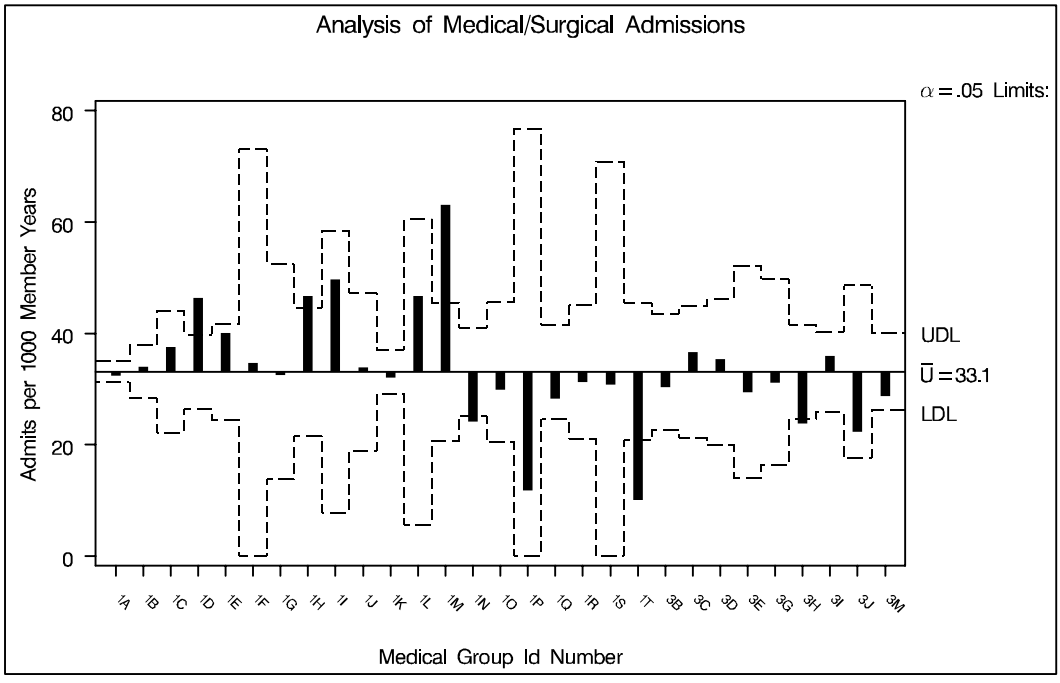
See ANMUEX1 in the SAS/QC Sample Library

Consider the example described in "Creating ANOM Charts for Rates from Group Counts." In the example, the option TURNHLABELS was used to vertically display the horizontal axis labels. You can also use an AXIS statement to create an angled display of the horizontal or vertical axes labels. The following statements create the u CHART shown in Output 4.1.1:

```
title 'Analysis of Medical/Surgical Admissions';
proc anom data=MSAdmits;
   uchart Count*ID / groupn   = KMemberYrs
                     cneedles = black
                     nolegend
                     haxis    = axis1;
axis1 value = (a=-45 h=2.0pct);
label Count = 'Admits per 1000 Member Years';
run;
```

The angle is specified with the a= option in the AXIS1 statement. Valid angle values are between -90 and 90. The height of the labels is specified with the h= option in the AXIS1 statement. See Axis and Axis Label Options on page 93.

Output 4.1.1. ANOM u Chart for C-Sections with Angled Axis Labels



Chapter 5
BOXCHART Statement

Chapter Contents

OVERVIEW 111

GETTING STARTED 111
- Creating ANOM Boxcharts from Response Values 112
- Creating ANOM Boxcharts from Group Summary Data 114
- Saving Summary Statistics for Groups 117
- Saving Decision Limits 118

SYNTAX 120
- Summary of Options 121

DETAILS 127
- Constructing ANOM Boxcharts 127
- Output Data Sets 130
- ODS Tables 133
- Input Data Sets 133
- Axis Labels 139
- Missing Values 139

EXAMPLES 139
- Example 5.1. ANOM Boxcharts with Unequal Group Sizes 139

Chapter 5
BOXCHART Statement

Overview

The BOXCHART statement creates an ANOM chart for group (treatment level) means of response values superimposed with box-and-whisker plots of the measurements in each group. Throughout this chapter, a chart of this type is referred to as an *ANOM boxchart*. You can use options in the BOXCHART statement to

- compute decision limits from the data based on a specified parameters, such as the significance level (α)
- tabulate group sample sizes, group means, decision limits, and other information
- save decision limits in an output data set
- save group sample sizes and group means in an output data set
- read decision limits and decision limit parameters from a data set
- display distinct sets of decision limits for different sets of groups
- specify one of several methods for calculating quantile statistics (percentiles)
- control the style of the box-and-whisker plots
- add block legends and symbol markers to identify special groups
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

Getting Started

This section introduces the BOXCHART statement with simple examples that illustrate the most commonly used options. Complete syntax for the BOXCHART statement is presented in the "Syntax" section on page 120, and advanced examples are given in the "Examples" section on page 139.

Creating ANOM Boxcharts from Response Values

See ANMBX1 in the SAS/QC Sample Library

A manufacturing engineer carries out a study to determine the source of excessive variation in the positioning of labels on shampoo bottles.* A labeling machine removes bottles from the line, attaches the labels, and returns the bottles to the line. There are six positions on the machine, and the engineer suspects that one or more of the position heads might be faulty.

A sample of 60 bottles, 10 per position, is run through the machine. For each bottle, the deviation of each label is measured in millimeters, and the machine position is recorded. The following statements create a SAS data set named LabelDeviations, which contains the deviation measurements for the 60 bottles:

```
data LabelDeviations;
   input Position @;
   do i = 1 to 5;
      input Deviation @;
      output;
   end;
   drop i;
   datalines;
1  -0.0239  -0.0285  -0.0300  -0.0043  -0.0362
1  -0.0422  -0.0014  -0.0647   0.0094  -0.0016
2  -0.0201  -0.0273   0.0227  -0.0332   0.0366
2   0.0438   0.0556   0.0098   0.0564   0.0182
3  -0.0073   0.0285  -0.0440  -0.0221  -0.0139
3   0.0486   0.0357   0.0235   0.0134  -0.0020
4   0.0669   0.1073   0.0597   0.0609   0.0755
4   0.0362   0.0561   0.0899   0.0418   0.0530
5   0.0368   0.0036   0.0374   0.0116  -0.0074
5   0.0250  -0.0080   0.0302  -0.0015  -0.0464
6   0.0049  -0.0384  -0.0204  -0.0049  -0.0120
6   0.0071  -0.0308   0.0017  -0.0285  -0.0070
run;
```

A partial listing of LabelDeviations is shown in Figure 5.1.

The Data Set LabelDeviations

Position	Deviation
1	-0.0239
1	-0.0285
1	-0.0300
1	-0.0043
1	-0.0362
1	-0.0422
1	-0.0014
1	-0.0647
1	0.0094
1	-0.0016

Figure 5.1. Listing of the Data Set LabelDeviations

*This example is based on a case study described by Hansen (1990).

The data set LabelDeviations is said to be in "strung-out" form, since each observation contains the position and the deviation measurement for a single bottle. The first 10 observations contain the measurements for the first position, the second 10 observations contain the measurements for the second position, and so on. Because the variable Position classifies the observations into groups (treatment levels), it is referred to as the *group-variable*. The variable Deviation contains the deviation measurements and is referred to as the *response variable* (or *response* for short).

The following statements create the ANOM boxchart shown in Figure 5.2:

```
title 'Analysis of Label Deviations';
proc anom data=LabelDeviations;
   boxchart Deviation*Position / alpha = 0.05;
   label Deviation = 'Mean Deviation from Center (mm)';
   label Position  = 'Labeler Position';
run;
```

This example illustrates the basic form of the BOXCHART statement. After the keyword BOXCHART, you specify the *response* to analyze (in this case, Deviation) followed by an asterisk and the *group-variable* (Position). Options are specified after the slash (/) in the BOXCHART statement. A complete list of options is presented in the "Syntax" section on page 120.

The input data set is specified with the DATA= option in the PROC ANOM statement when it contains raw measurements for the *response*.

Each point on the ANOM chart represents the average (mean) of the response measurements for a particular sample.

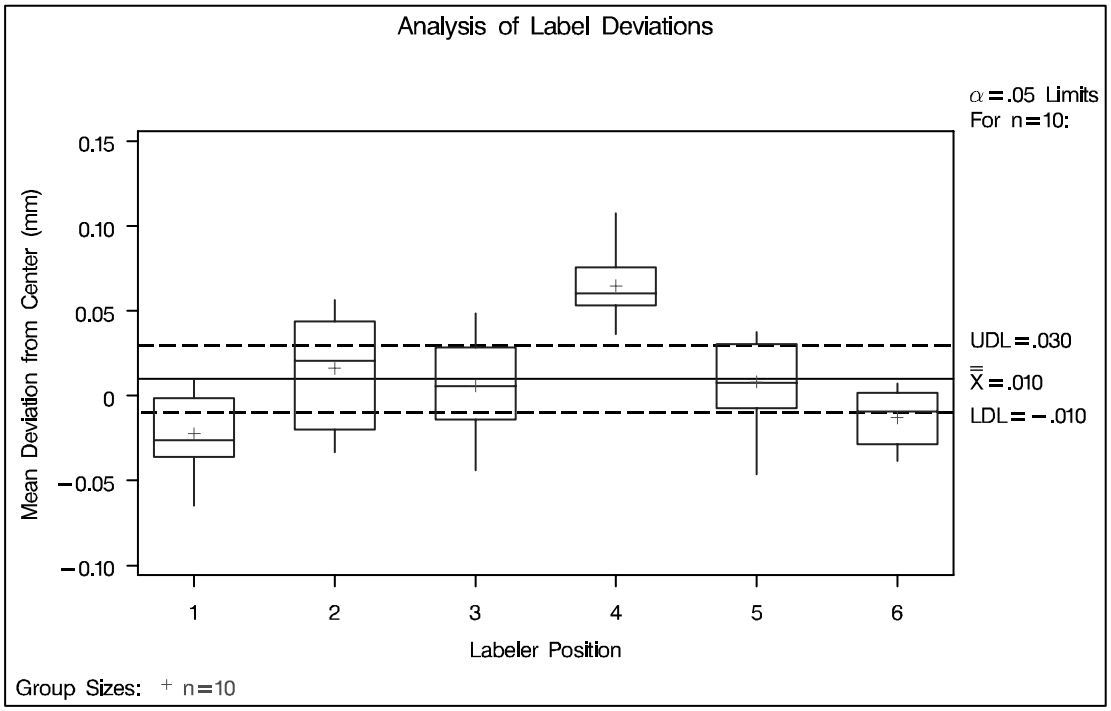


Figure 5.2. ANOM Chart for Means of Labeler Position Data

The average for Position 1 is below the lower decision limit (LDL), and the average for Position 6 is slightly below the lower decision limit. The average for Position 4 exceeds the upper decision limit (UDL). The conclusion is that Positions 1, 4, and 6 are operating differently.

By default, the decision limits shown correspond to a significance level of $\alpha = 0.05$; the formulas for the limits are given in the section "Decision Limits" on page 128. You can also read decision limits from an input data set.

For computational details, see "Constructing ANOM Boxcharts" on page 127. For details on reading raw measurements, see "DATA= Data Set" on page 135.

Creating ANOM Boxcharts from Group Summary Data

See ANMBXGRP in the SAS/QC Sample Library

The previous example illustrates how you can create ANOM charts for means using measurement data. However, in many applications, the data are provided as group summary statistics. This example illustrates how you can use the BOXCHART statement with data of this type.

The following data set (Labels) provides the data from the preceding example in summarized form:

```
data Labels;
   input Position DeviationL Deviation1 DeviationX
         DeviationM Deviation3 DeviationH DeviationS;
   DeviationN = 10;
   datalines;
1  -0.0647  -0.0362  -0.02234  -0.02620  -0.0016  0.0094  0.02281
2  -0.0332  -0.0201   0.01625   0.02045   0.0438  0.0564  0.03347
3  -0.0440  -0.0139   0.00604   0.00570   0.0285  0.0486  0.02885
4   0.0362   0.0530   0.06473   0.06030   0.0755  0.1073  0.02150
5  -0.0464  -0.0074   0.00813   0.00760   0.0302  0.0374  0.02593
6  -0.0384  -0.0285  -0.01283  -0.00950   0.0017  0.0071  0.01599
run;
```

A listing of Labels is shown in Figure 5.3. There is exactly one observation for each group (note that the groups are still indexed by Position). There are eight summary variables in Labels.

- DeviationL contains the group minimums (low values).
- Deviation1 contains the 25th percentile (first quartile) of each group.
- DeviationX contains the group means.
- DeviationM contains the group medians.
- Deviation3 contains the 75th percentile (third quartile) of each group.
- DeviationH contains the group maximums (high values).
- DeviationS contains the group standard deviations.
- DeviationN contains the group sample sizes (these are all 10 in this case).

The Data Set Labels

Position	Deviation L	Deviation 1	Deviation X	Deviation M	Deviation 3	Deviation H	Deviation S	Deviation N
1	-0.0647	-0.0362	-0.02234	-0.02620	-0.0016	0.0094	0.02281	10
2	-0.0332	-0.0201	0.01625	0.02045	0.0438	0.0564	0.03347	10
3	-0.0440	-0.0139	0.00604	0.00570	0.0285	0.0486	0.02885	10
4	0.0362	0.0530	0.06473	0.06030	0.0755	0.1073	0.02150	10
5	-0.0464	-0.0074	0.00813	0.00760	0.0302	0.0374	0.02593	10
6	-0.0384	-0.0285	-0.01283	-0.00950	0.0017	0.0071	0.01599	10

Figure 5.3. The Summary Data Set Labels

You can read this data set by specifying it as a SUMMARY= data set in the PROC ANOM statement, as follows:

```
title 'Analysis of Label Deviations';
proc anom summary=Labels;
   boxchart Deviation*Position;
run;
```

The resulting ANOM boxchart is shown in Figure 5.4. Note that Deviation is *not* the name of a SAS variable in the data set but is, instead, the common prefix for the names of the eight summary variables. The suffix characters *L*, *1*, *X*, *M*, *3*, *H*, *S*, and *N* indicate the contents of the variable. For example, the suffix characters *1* and *3* indicate first and third quartiles. Thus, you can specify three group summary variables in a SUMMARY= data set with a single name (Deviation), which is referred to as the *response*. The name Position specified after the asterisk is the name of the *group-variable*.

In general, a SUMMARY= input data set used with the BOXCHART statement must contain the following variables:

- group variable
- group minimum variable
- group first quartile variable
- group mean variable
- group median variable
- group third quartile variable
- group maximum variable
- group standard deviation variable
- group sample size variable

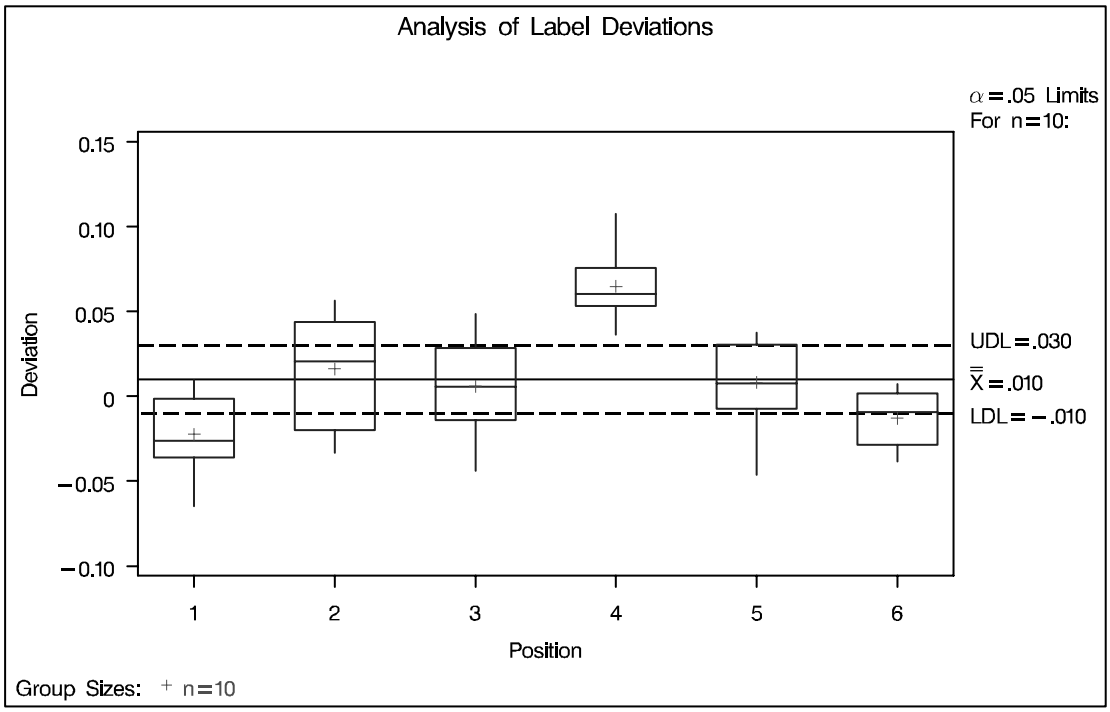


Figure 5.4. ANOM Chart for Means in Data Set Labels

Furthermore, the names of the summary variables must begin with the *response* name specified in the BOXCHART statement and end with the appropriate suffix characters. If the names do not follow this convention, you can use the RENAME option in the PROC ANOM statement to rename the variables for the duration of the ANOM procedure step. If a label is associated with the group mean variable, it is used to label the vertical axis.

In summary, the interpretation of *response* depends on the input data set.

- If raw data are read using the DATA= option (as in the previous example), *response* is the name of the SAS variable containing the response measurements.
- If summary data are read using the SUMMARY= option (as in this example), *response* is the common prefix for the names of the variables containing the summary statistics.

For more information, see "SUMMARY= Data Set" on page 136.

Saving Summary Statistics for Groups

See ANMBXSUM in the SAS/QC Sample Library

In this example, the BOXCHART statement is used to create a data set containing group summary statistics that can be read later by the ANOM procedure (as in the preceding example). The following statements read measurements from the data set LabelDeviations and create a summary data set named LabelSummary:

```
proc anom data=LabelDeviations;
   boxchart Deviation*Position / outsummary=LabelSummary
                                nochart;
run;
```

The OUTSUMMARY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to Figure 5.2.

Figure 5.5 contains a listing of LabelSummary.

The Data Set LabelSummary

Position	DeviationL	Deviation1	DeviationX	DeviationM	Deviation3	DeviationH	DeviationS	DeviationN
1	-0.0647	-0.0362	-0.02234	-0.02620	-0.0016	0.0094	0.022807	10
2	-0.0332	-0.0201	0.01625	0.02045	0.0438	0.0564	0.033473	10
3	-0.0440	-0.0139	0.00604	0.00570	0.0285	0.0486	0.028849	10
4	0.0362	0.0530	0.06473	0.06030	0.0755	0.1073	0.021495	10
5	-0.0464	-0.0074	0.00813	0.00760	0.0302	0.0374	0.025928	10
6	-0.0384	-0.0285	-0.01283	-0.00950	0.0017	0.0071	0.015986	10

Figure 5.5. The Summary Data Set LabelSummary

There are nine variables in the data set LabelSummary.

- Position identifies the group.
- DeviationL contains the group minimums.
- Deviation1 contains the first quartile for each group.
- DeviationX contains the group means.
- DeviationM contains the group medians.
- Deviation3 contains the third quartile for each group.
- DeviationH contains the group maximums.
- DeviationS contains the group standard deviations.
- DeviationN contains the group sizes.

Note that the summary statistic variables are named by adding the suffix characters *L*, *1*, *X*, *M*, *3*, *H*, *S*, and *N* to the *response* Deviation specified in the BOXCHART statement. In other words, the variable naming convention for OUTSUMMARY= data sets is the same as that for SUMMARY= data sets.

For more information, see "OUTSUMMARY= Data Set" on page 131.

Saving Decision Limits

See ANMBXLIM in the SAS/QC Sample Library

You can save the decision limits for an ANOM chart, together with the parameters used to compute the limits, in a SAS data set.

The following statements read measurements from the data set LabelDeviations (see "Creating ANOM Boxcharts from Response Values" on page 112.) and save the decision limits displayed in Figure 5.2 in a data set named LabelLimits:

```
proc anom data=LabelDeviations;
   boxchart Deviation*Position / outlimits=LabelLimits
                                nochart;
run;
```

The OUTLIMITS= option names the data set containing the decision limits, and the NOCHART option suppresses the display of the chart. The data set LabelLimits is listed in Figure 5.6.

Decision Limits for Labler Position Deviations

VAR	_GROUP_	_TYPE_	_LIMITK_	_ALPHA_	_LDLX_	_MEAN_	_UDLX_	_LIMITN_	_MSE_	_DFE_	_LIMITM_
Deviation	Position	ESTIMATE	10	0.05	-.009875608	.009996667	0.029869	.000643787	54	6	

Figure 5.6. The Data Set LabelLimits Containing Decision Limit Information

The data set LabelLimits contains one observation with the limits for *response* Deviation. The values of _LDLX_ and _UDLX_ are the lower and upper decision limits for the means, and the value of _MEAN_ is the weighted average of the group means, which is represented by the central line.

The values of _MEAN_, _MSE_, _DFE_,_LIMITK_, _LIMITN_, and _ALPHA_ are the parameters used to compute the decision limits. The value of _MSE_ is the mean square error, and the value of _DFE_ is the associated degrees of freedom. The value of _LIMITK_ is the group size (k), the value of _LIMITN_ is the nominal sample size associated with the decision limits, and the value of _ALPHA_ is the value of the significance level (α). The variables _VAR_ and _GROUP_ are bookkeeping variables that save the *response* and *group-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of

MEAN and _MSE_ are estimates computed from the data or standard (known) values specified with procedure options. In most applications, the value of _TYPE_ will be ESTIMATE.

You can create an output data set containing both decision limits and group summary statistics with the OUTTABLE= option, as illustrated by the following statements:

See ANMBXTAB in the SAS/QC Sample Library

```
proc anom data=LabelDeviations;
   boxchart Deviation*Position / outtable=LabelTab
                                nochart;
run;
```

The data set LabelTab is listed in Figure 5.7.

Summary Statistics and Decision Limits

VAR	Position	_ALPHA_	_LIMITN_	_SUBN_	_LDLX_	_SUBX_	_MEAN_
Deviation	1	0.05	10	10	-.009875608	-0.02234	.009996667
Deviation	2	0.05	10	10	-.009875608	0.01625	.009996667
Deviation	3	0.05	10	10	-.009875608	0.00604	.009996667
Deviation	4	0.05	10	10	-.009875608	0.06473	.009996667
Deviation	5	0.05	10	10	-.009875608	0.00813	.009996667
Deviation	6	0.05	10	10	-.009875608	-0.01283	.009996667

UDLX	_EXLIM_	_SUBMIN_	_SUBQ1_	_SUBMED_	_SUBQ3_	_SUBMAX_
0.029869	LOWER	-0.0647	-0.0362	-0.02620	-0.0016	0.0094
0.029869		-0.0332	-0.0201	0.02045	0.0438	0.0564
0.029869		-0.0440	-0.0139	0.00570	0.0285	0.0486
0.029869	UPPER	0.0362	0.0530	0.06030	0.0755	0.1073
0.029869		-0.0464	-0.0074	0.00760	0.0302	0.0374
0.029869	LOWER	-0.0384	-0.0285	-0.00950	0.0017	0.0071

Figure 5.7. The Data Set LabelTab

This data set contains one observation for each group sample. The variable _SUBMIN_ contains the group minimums, and the variable _SUBQ1_ contains the first quartile for each group. The variables _SUBX_ and _SUBMED_ contain the group means and medians. The variable _SUBQ3_ contains the third quartiles, _SUBMAX_ contains the group maximums, and _SUBN_ contains the group sample sizes. The variables _LDLX_ and _UDLX_ contain the lower and upper decision limits, and the variable _MEAN_ contains the central line. The variables _VAR_ and Position contain the *response* name and values of the *group-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 132.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read LabelTab and display an ANOM boxchart (not shown here) identical to the chart in Figure 5.2:

```
title 'Analysis of Label Deviations';
proc anom table=LabelTab;
   xchart deviation*position;
label _SUBX_ = 'Mean Deviation from Center (mm)';
run;
```

Because the ANOM procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized ANOM boxcharts.

For more information, see "TABLE= Data Set" on page 138.

Syntax

The basic syntax for the BOXCHART statement is as follows:

BOXCHART *response * group-variable* **;**

The general form of this syntax is as follows:

BOXCHART *response * group-variable* < *(block-variables)* > < *=symbol-variable* > < *options* >*;* **;**

You can use any number of BOXCHART statements in the ANOM procedure. The components of the BOXCHART statement are described as follows.

responses
identify one or more responses to be analyzed. The specification of *response* depends on the input data set specified in the PROC ANOM statement.

- If response values (raw data) are read from a DATA= data set, *response* must be the name of the variable containing the values.For an example, see "Creating ANOM Boxcharts from Response Values" on page 112.
- If summary data are read from a SUMMARY= data set, *response* must be the common prefix of the summary variables in the SUMMARY= data set. For an example, see "Creating ANOM Boxcharts from Group Summary Data" on page 114.
- If summary data and decision limits are read from a TABLE= data set, *response* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Decision Limits" on page 118.

A *response* is required. If you specify more than one response, enclose the list in parentheses. For example, the following statements request distinct ANOM charts for the means of WEIGHT, LENGTH, and WIDTH:

```
proc anom data=measures;
   xchart (weight length width)*day;
run;
```

group-variable
is the variable that identifies groups in the data. The *group-variable* is required. In the preceding BOXCHART statement, DAY is the group variable.

block-variables
are optional variables that group the data into blocks of consecutive groups. The

blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker used to plot the means. Distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOLn statements.

options
enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The "Summary of Options" section, which follows, lists all options by function.

Summary of Options

The following tables list the BOXCHART statement options by function. Many of these options are identical to options in the SHEWHART procedure, which are described in detail beginning on page 1851 in Chapter 53, "Dictionary of Options."

Table 5.1. Tabulation Options

TABLE	creates a basic table of group means, group sample sizes, and decision limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, and TABLEOUTLIM
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLEOUTLIM	augments basic table with columns indicating decision limits exceeded

Table 5.2. Options for Controlling Box Appearance

BOXCONNECT	connects group means in box-and-whisker plots
BOXCONNECT=*keyword*	connects group means, medians, maximum values, minimum values, or quartiles in box-and-whisker plots
BOXSTYLE=*keyword*	specifies style of box-and-whisker plots
BOXWIDTH=*value*	specifies width of box-and-whisker plots
BOXWIDTHSCALE=*value*	specifies that widths of box-and-whisker plots vary proportionately to group sample size
CBOXES=*color*|(*variable*)	specifies color for outlines of box-and-whisker plots
CBOXFILL=*color*|(*variable*)	specifies fill color for interior of box-and-whisker plots
IDCOLOR=*color*	specifies outlier symbol color in schematic box-and-whisker plots
IDCTEXT=*color*	specifies text color to label outliers or response variable values
IDFONT=*font*	specifies text font to label outliers or response variable values
IDHEIGHT=*value*	specifies text height to label outliers or response variable values
IDSYMBOL=*symbol*	specifies outlier symbol in schematic box-and-whisker plots
LBOXES=*linetype*|(*variable*)	specifies line types for outlines of box-and-whisker plots
NOTCHES	specifies that box-and-whisker plots are to be notched
PCTLDEF=*n*	specifies percentile definition used for box-and-whisker plots
SERIFS	adds serifs to the whiskers of skeletal box-and-whisker plots

Note that specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only, that is, groups for which the mean exceeds the decision limits.

Table 5.3. Reference Line Options

CHREF=*color*	specifies color for lines requested by HREF= option
CVREF=*color*	specifies color for lines requested by VREF= option
HREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on ANOM boxchart
HREFDATA= *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on ANOM boxchart
HREFLABELS= *'label1'...'labeln'*	specifies labels for HREF= lines
HREFLABPOS=*n*	specifies position of HREFLABELS= labels
LHREF=*linetype*	specifies line type for HREF= lines
LVREF=*linetype*	specifies line type for VREF= lines
NOBYREF	specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to vertical axis on ANOM boxchart
VREFLABELS= *'label1'...'labeln'*	specifies labels for VREF= lines
VREFLABPOS=*n*	position of VREFLABELS= labels

Table 5.4. Block Variable Legend Options

BLOCKLABELPOS= *keyword*	specifies position of label for *block-variable* legend
BLOCKLABTYPE= *value*|*keyword*	specifies text size of *block-variable* legend
BLOCKPOS=*n*	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=*color*	specifies color for filling background in *block-variable* legend
CBLOCKVAR=*variable*| (*variables*)	specifies one or more variables whose values are colors for filling background of *block-variable* legend

Table 5.5. Axis and Axis Label Options

CAXIS=*color*	specifies color for axis lines and tick marks
CFRAME=*color*| (*color-list*)	specifies fill colors for frame for plot area
CTEXT=*color*	specifies color for tick mark values and axis labels
HAXIS=*values*|AXISn	specifies major tick mark values for horizontal axis
HEIGHT=*value*	specifies height of axis label and axis legend text
HMINOR=*n*	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=*value*	specifies length of offset at both ends of horizontal axis
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent group values is to be labeled on horizontal axis
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=*n*	specifies thinning factor for tick mark labels on horizontal axis
SPLIT='*character*'	specifies splitting character for axis labels
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=*values*|AXISn	specifies major tick mark values for vertical axis of ANOM box-chart
VMINOR=*n*	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=*value*	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis for ANOM boxchart
WAXIS=*n*	specifies width of axis lines

Table 5.6. Options for Specifying Parameters for Decision Limits

ALPHA=*value*	specifies the probability of a Type I error
DFE=*number*	specifies the degrees of freedom associated with the root mean square error
LIMITK=*k*	specifies number of groups for decision limits
LIMITN=*n*|VARYING	specifies either a nominal sample size for fixed decision limits or varying limits
MEAN=*value*	specifies the mean
NOREADLIMITS	computes decision limits for each *response* from the data rather than a LIMITS= data set
READINDEXES=ALL| '*label1*'...'*labeln*'	reads multiple sets of decision limits for each *response* from a LIMITS= data set
MSE=*value*	specifies the mean square error
TYPE=*keyword*	identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set

Table 5.7. Options for Displaying Decision Limits

CINFILL=*color*	specifies color for area inside decision limits
CLIMITS=*color*	specifies color of decision limits, central line, and related labels
LDLLABEL='*label*'	specifies label for lower decision limit
LIMLABSUBCHAR='*character*'	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the decision limit
LLIMITS=*linetype*	specifies line type for decision limits
NDECIMAL=*n*	specifies number of digits to right of decimal place in default labels for decision limits and central line
NOCTL	suppresses display of central line
NOLDL	suppresses display of lower decision limit
NOLIMITLABEL	suppresses labels for decision limits and central line
NOLIMITS	suppresses display of decision limits
NOLIMITSFRAME	suppresses default frame around decision limit information when multiple sets of decision limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for decision limits
NOUDL	suppresses display of upper decision limit
UDLLABEL='*string*'	specifies label for upper decision limit
WLIMITS=*n*	specifies width for decision limits and central line
XSYMBOL='*string*'|*keyword*	specifies label for central line

Table 5.8. Options for Plotting and Labeling Points

ALLLABEL=VALUE|(*variable*)	labels every point on ANOM chart
CCONNECT=*color*	specifies color for line segments connecting points on chart
CFRAMELAB=*color*	specifies fill color for frame around labeled points
CNEEDLES=*color*	specifies color for needles that connect points to central line
COUT=*color*	specifies color for portions of line segments that connect points outside decision limits
COUTFILL=*color*	specifies color for shading areas between connected points and decision limits outside the limits
NONEEDLES	suppresses vertical needles connecting points to central line
OUTLABEL=VALUE|(*variable*)	labels points outside decision limits
SYMBOLLEGEND=NONE|*name*	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=*keyword*	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=*n*	specifies width of needles

Table 5.9. Clipping Options

CCLIP=*color*	specifies color for plot symbol for clipped points
CLIPFACTOR=*value*	determines extent to which extreme points are clipped
CLIPLEGEND='*string*'	specifies text for clipping legend
CLIPLEGPOS=*keyword*	specifies position of clipping legend
CLIPSUBCHAR='*character*'	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=*symbol*	specifies plot symbol for clipped points
CLIPSYMBOLHT=*value*	specifies symbol marker height for clipped points

Table 5.10. Phase Options

CPHASELEG=*color*	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE='*string*'	specifies value of _PHASE_ in the OUTSUMMARY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=*value*|*keyword*	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels decision limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES= ALL|'*label1*'...'*labeln*'	specifies *phases* to be read from an input data set

Table 5.11. Options for Interactive ANOM Charts

HTML=(*variable*)	specifies a variable whose values are URLs to be associated with groups
HTML_LEGEND=(*variable*)	specifies a variable whose values are URLs to be associated with symbols in the symbol legend
WEBOUT=*SAS-data-set*	creates an OUTTABLE= data set with additional graphics coordinate data

Table 5.12. Output Data Set Options

OUTBOX=*SAS-data-set*	creates output data set containing group summary statistics, decision limits, and outlier values
OUTINDEX='*string*'	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=*SAS-data-set*	creates output data set containing decision limits
OUTSUMMARY=*SAS-data-set*	creates output data set containing group summary statistics
OUTTABLE=*SAS-data-set*	creates output data set containing group summary statistics and decision limits

Table 5.13. Grid Options

GRID	adds grid to ANOM chart
LGRID=*linetype*	specifies line type for grid requested with the GRID option
WGRID=*n*	specifies width of grid lines

Table 5.14. Plot Layout Options

ALLN	plots means for all groups
BILEVEL	creates ANOM boxcharts using half-screens and half-pages
EXCHART	creates ANOM boxcharts for a response only when a group mean exceeds the decision limits
MAXPANELS=*n*	maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed decision limits
NOCHART	suppresses creation of chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for group sample sizes
NPANELPOS=*n*	specifies number of group positions per panel on each chart
REPEAT	repeats last group position on panel as first group position of next panel
TOTPANELS=*n*	specifies number of pages or screens to be used to display chart
ZEROSTD	displays ANOM boxchart regardless of whether the root mean square error is zero

Table 5.15. Graphical Enhancement Options

ANNOTATE= *SAS-data-set*	specifies annotate data set that adds features to ANOM boxchart
DESCRIPTION='*string*'	specifies string that appears in the description field of the PROC GREPLAY master menu for ANOM boxchart
FONT=*font*	specifies software font for labels and legends on charts
NAME='*string*'	specifies name that appears in the name field of the PROC GREPLAY master menu for ANOM boxchart
PAGENUM='*string*'	specifies the form of the label used in pagination
PAGENUMPOS= *keyword*	specifies the position of the page number requested with the PAGENUM= option

Table 5.16. Overlay Options

CCOVERLAY=(*color-list*)	specifies colors for ANOM chart overlay line segments
COVERLAY=(*color-list*)	specifies colors for ANOM chart overlay plots
COVERLAYCLIP=*color*	specifies color for clipped points on overlays
LOVERLAY=(*linetypes*)	specifies line types for ANOM chart overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=(*variable-list*)	specifies variables to overlay on ANOM chart
OVERLAYCLIPSYM=*symbol*	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=*value*	specifies symbol height for clipped points on overlays
OVERLAYHTML=(*variable-list*)	specifies URLS to associate with ANOM chart overlay points
OVERLAYLEGLAB='*label*'	specifies label for overlay legend
OVERLAYSYM=(*symbol-list*)	specifies symbols for ANOM chart overlays
OVERLAYSYMHT=(*value-list*)	specifies symbol heights for ANOM chart overlays
WOVERLAY=(*value-list*)	specifies widths of ANOM chart overlay line segments

Details

Constructing ANOM Boxcharts

The following notation is used in this section:

X_{ij}	jth response in the ith group
k	number of groups
n_i	sample size of ith group
N	total sample size $= n_1 + \cdots + n_k$
μ_i	expected value of the responses in the ith group
σ	standard deviation of the responses in the ith group
$\bar{X}_i$	mean of responses in ith group
$\overline{\overline{X}}$	weighted average of k group means
s_i^2	variance of responses in ith group
$\widehat{\sigma^2}$	mean square error (MSE)
ν	degrees of freedom associated with the mean square error
α	significance level
$h(\alpha; k, n, \nu)$	critical value for analysis of means when the sample sizes n_i are equal ($n_i \equiv n$)
$h(\alpha; k, n_1, \ldots, n_k, \nu)$	critical value for analysis of means when the sample sizes n_i are not equal

Elements of Box-and-Whisker Plots

A box-and-whisker plot is displayed for the measurements in each group on the ANOM boxchart. Figure 5.8 illustrates the elements of each plot.

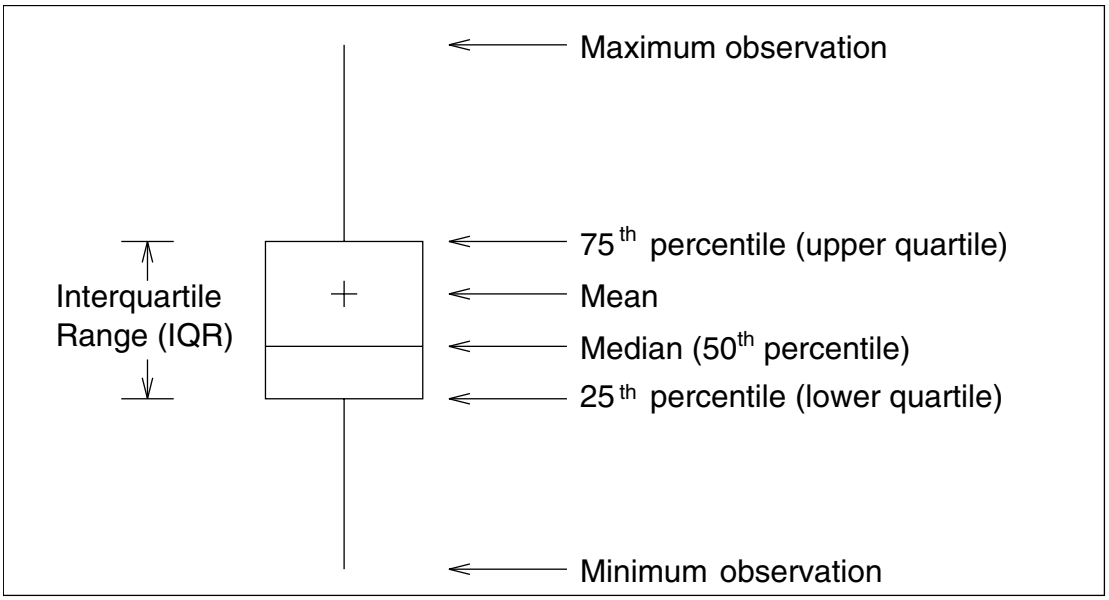


Figure 5.8. Box-and-Whisker Plot

The skeletal style of the box-and-whisker plot shown in Figure 5.8 is the default. You can specify alternative styles with the BOXSTYLE= option; see the entry for the BOXSTYLE= option in Chapter 53, "Dictionary of Options."

Central Line

By default, the central line on an ANOM chart for means represents the weighted average of the group means, which is computed as

$$\overline{\overline{X}} = \frac{n_1\bar{X}_1 + \cdots + n_k\bar{X}_k}{n_1 + \cdots + n_k}$$

You can specify a value for $\overline{\overline{X}}$ with the MEAN= option in the BOXCHART statement or with the variable _MEAN_ in a LIMITS= data set.

Decision Limits

In the analysis of means for continuous data, it is assumed that the responses in the ith group are at least approximately normally distributed with a constant variance:

$$X_{ij} \sim N(\mu_i, \sigma^2), \qquad j = 1, \ldots, n_i$$

When the group sizes are constant ($n_i \equiv n$), then $\nu = N - k = k(n-1)$ and the decision limits are computed as follows:

$$\begin{aligned}\text{lower decision limit (LDL)} &= \overline{\overline{X}} - h(\alpha; k, n, \nu)\sqrt{\text{MSE}}\sqrt{\frac{k-1}{N}} \\ \text{upper decision limit (UDL)} &= \overline{\overline{X}} + h(\alpha; k, n, \nu)\sqrt{\text{MSE}}\sqrt{\frac{k-1}{N}}\end{aligned}$$

Here the mean square error (MSE) is computed as follows:

$$\text{MSE} = \widehat{\sigma^2} = \frac{1}{k}\sum_{j=1}^{k} s_j^2$$

For details concerning the function $h(\alpha; k, n, \nu)$, see Nelson (1981, 1982a, 1993).

When the group sizes n_i are not constant (the unbalanced case), $\nu = N - k$ and the decision limits for the ith group are computed as follows:

$$\begin{aligned}\text{lower decision limit (LDL)} &= \overline{\overline{X}} - h(\alpha; k, n_1, \ldots, n_k, \nu)\sqrt{\text{MSE}}\sqrt{\frac{N-n_i}{Nn_i}} \\ \text{upper decision limit (UDL)} &= \overline{\overline{X}} + h(\alpha; k, n_1, \ldots, n_k, \nu)\sqrt{\text{MSE}}\sqrt{\frac{N-n_i}{Nn_i}}\end{aligned}$$

Here the mean square error (MSE) is computed as follows:

$$\text{MSE} = \widehat{\sigma^2} = \frac{(n_1-1)s_1^2 + \cdots + (n_k-1)s_k^2}{n_1 + \cdots + n_k - k}$$

This requires that ν be positive. A chart is not produced if $\nu > 0$ but MSE is equal to zero (unless you specify the ZEROSTD option). For details concerning the function $h(\alpha; k, n_1, \ldots, n_k, \nu)$, see Fritsch and Hsu (1997), Nelson (1982b, 1991), and Soong and Hsu (1997).

You can specify parameters for the limits as follows:

- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set. By default, α = 0.05.
- Specify a constant nominal sample size $n_i \equiv n$ for the decision limits in the balanced case with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set. By default, n is the observed sample size in the balanced case.
- Specify k with the LIMITK= option or with the variable _LIMITK_ in a LIMITS= data set. By default, k is the number of groups.
- Specify $\overline{\overline{X}}$ with the MEAN= option or with the variable _MEAN_ in a LIMITS= data set. By default, $\overline{\overline{X}}$ is the weighted average of the responses.

- Specify $\widehat{\sigma^2}$ with the MSE= option or with the variable _MSE_ in a LIMITS= data set. By default, $\widehat{\sigma^2}$ is computed as indicated above.
- Specify ν with the DFE= option or with the variable _DFE_ in a LIMITS= data set. By default, ν is determined as indicated above.

Output Data Sets

OUTBOX= Data Set

The OUTBOX= data set saves group summary statistics, decision limits, and outlier values. The following variables can be saved:

- the *group-variable*
- the variable _VAR_, containing the analysis variable name
- the variable _TYPE_, identifying features of box-and-whisker plots
- the variable _VALUE_, containing values of box-and-whisker plot features
- the variable _ID_, containing labels for outliers
- the variable _HTML_, containing URLs associated with box-and-whisker plot features

_ID_ is included in the OUTBOX= data set only if one of the keywords SCHEMATICID or SCHEMATICIDFAR is specified with the BOXSTYLE= option. _HTML_ is present only if one or more of the HTML=, OUTHIGHHTML=, OUTLOWHTML=, or POINTSHTML= options are specified.

Each observation in an OUTBOX= data set records the value of a single feature of one group's box-and-whisker plot, such as its mean. The _TYPE_ variable identifies the feature whose value is recorded in _VALUE_. The following table lists valid _TYPE_ variable values:

Table 5.17. Valid _TYPE_ Values in an OUTBOX= Data Set

TYPE Value	Description
N	group size
ALPHA	significance level
LIMITN	nominal sample size associated with decision limits
LDLX	lower decision limit for group mean
UDLX	upper decision limit for group mean
RESPMEAN	overall response variable mean
MIN	group minimum value
Q1	group first quartile
MEDIAN	group median
MEAN	group mean
Q3	group third quartile
MAX	group maximum value
LOW	low outlier value
HIGH	high outlier value
LOWHISKR	low whisker value, if different from MIN
HIWHISKR	high whisker value, if different from MAX
FARLOW	low far outlier value
FARHIGH	high far outlier value

Additionally, the following variables, if specified, are included:

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

OUTLIMITS= Data Set

The OUTLIMITS= data set saves decision limits and decision limit parameters. The following variables can be saved:

Table 5.18. OUTLIMITS= Data Set

Variable	Description
ALPHA	significance level
DFE	degrees of freedom for mean square error
GROUP	*group-variable* specified in the BOXCHART statement
INDEX	optional identifier for the decision limits specified with the OUTINDEX= option
LDLX	lower decision limit for group means
LIMITK	number of groups
LIMITN	sample size associated with the decision limits
MEAN	weighted average of group means ($\overline{\overline{X}}$)
MSE	mean square error
TYPE	type (estimate or standard value) of _MEAN_ and _MSE_
UDLX	upper decision limit for group means
VAR	*response* specified in the BOXCHART statement

Notes:

1. In the unbalanced case, the special missing value *V* is assigned to the variables _LIMITN_, _LDLX_, and _UDLX_.
2. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *response* specified in the BOXCHART statement. For an example, see "Saving Decision Limits" on page 118.

OUTSUMMARY= Data Set

The OUTSUMMARY= data set saves group summary statistics. The following variables can be saved:

- the *group-variable*
- a group minimum variable named by *response* suffixed with *L*
- a group first-quartile variable named by *response* suffixed with *1*
- a group mean variable named by *response* suffixed with *X*
- a group median variable named by *response* suffixed with *M*

- a group third-quartile variable named by *response* suffixed with *3*
- a group maximum variable named by *response* suffixed with *H*
- a group standard deviation variable named by *response* suffixed with *S*
- a group sample size variable named by *response* suffixed with *N*

Given a *response* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Group summary variables are created for each *response* specified in the BOXCHART statement. For example, consider the following statements:

```
proc anom data=Steel;
   xchart (Width Diameter)*lot / outsummary=Summary;
run;
```

The data set Summary contains variables named lot, WidthL, Width1, WidthX, WidthM, Width3, WidthH, WidthS, WidthN, DiameterL, Diameter1, DiameterX, DiameterM, Diameter3, DiameterH, DiameterS, and DiameterN. Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTSUMMARY= data set, see "Saving Summary Statistics for Groups" on page 117.

OUTTABLE= Data Set

The OUTTABLE= data set saves group summary statistics, decision limits, and related information. The following variables can be saved:

Variable	Description
ALPHA	significance level
EXLIM	decision limit exceeded (if any)
group	values of the group variable
LDLX	lower decision limit for group mean
LIMITN	nominal sample size associated with the decision limits
MEAN	central line
SUBMAX	group maximum
SUBMED	group median
SUBMIN	group minimum
SUBN	group sample size
SUBQ1	group first quartile
SUBQ3	group third quartile
SUBX	group mean
UDLX	upper decision limit for group mean
VAR	*response* specified in the BOXCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Note: The variable _EXLIM_ is a character variable of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Decision Limits" on page 118.

ODS Tables

The following table summarizes the ODS tables that you can request with the BOXCHART statement.

Table 5.19. ODS Tables Produced with the BOXCHART Statement

Table Name	Description	Options
BOXCHART	ANOM chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLEOUT

Input Data Sets

BOX= Data Set

You can read summary statistics, decision limits, and outlier values from a BOX= data set specified in the PROC ANOM statement. This enables you to reuse an OUTBOX= data set created in a previous run of the ANOM procedure to display a box chart.

A BOX= data set must contain the following variables:

- the group variable
- _VAR_, containing the analysis variable name
- _TYPE_, identifying features of box-and-whisker plots
- _VALUE_, containing values of those features

Each observation in a BOX= data set records the value of a single feature of one group's box-and-whisker plot, such as its mean. The _TYPE_ variable identifies the feature whose value is recorded in a given observation. The following table lists valid _TYPE_ variable values:

Table 5.20. Valid _TYPE_ Values in a BOX= Data Set

TYPE Value	Description
N	group size
ALPHA	significance level
LIMITN	nominal sample size associated with decision limits
LDLX	lower decision limit for group mean
UDLX	upper decision limit for group mean
RESPMEAN	overall response variable mean
MIN	group minimum value
Q1	group first quartile
MEDIAN	group median
MEAN	group mean
Q3	group third quartile
MAX	group maximum value
LOW	low outlier value
HIGH	high outlier value
LOWHISKR	low whisker value, if different from MIN
HIWHISKR	high whisker value, if different from MAX
FARLOW	low far outlier value
FARHIGH	high far outlier value

The features identified by _TYPE_ values N, LDLX, UDLX, RESPMEAN, MIN, Q1, MEDIAN, MEAN, Q3, and MAX are required for each group.

Other variables that can be read from a BOX= data set include:

- the variable _ID_, containing labels for outliers
- the variable _HTML_, containing URLs to be associated with features on box plots
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

When you specify one of the keywords SCHEMATICID or SCHEMATICIDFAR with the BOXSTYLE= option, values of _ID_ are used as outlier labels. If _ID_

does not exist in the BOX= data set, the values of the first variable listed in the ID statement are used.

DATA= Data Set

You can read raw data (response values) from a DATA= data set specified in the PROC ANOM statement. Each *response* specified in the BOXCHART statement must be a SAS variable in the DATA= data set. This variable provides measurements that must be grouped into group samples indexed by the *group-variable*. The *group-variable*, which is specified in the BOXCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *response* and a value for the *group-variable*. If the ith group contains n_i items, there should be n_i consecutive observations for which the value of the *group-variable* is the index of the ith group. For example, if each group contains five items and there are 10 groups, the DATA= data set should contain 50 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the ANOM procedure reads all of the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option.

For an example of a DATA= data set, see "Creating ANOM Boxcharts from Response Values" on page 112.

LIMITS= Data Set

You can read preestablished decision limits (or parameters from which the decision limits can be calculated) from a LIMITS= data set specified in the PROC ANOM statement. For example, the following statements read decision limit information from the data set conlims:

```
proc anom data=info limits=conlims;
   xchart weight*batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the ANOM procedure. Such data sets always contain the variables required for a LIMITS= data set; see Table 5.18. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LDLX_, _MEAN_, and _UDLX_, which specify the decision limits directly
- the variables _MEAN_, _MSE_, and _DFE_, which are used to calculate the decision limits according to the equations in the section "Decision Limits" on page 128.

In addition, note the following:

- The variables _VAR_ and _GROUP_ are required. These must be character variables whose lengths are no greater than 32.
- _DFE_ is optional. The default is $\nu = N - k$, and in the case of equal group sizes, $\nu = k(n-1)$.
- _MSE_ is optional if _LDLX_ and _UDLX_ are specified; otherwise it is required.
- _LDLX_ and _UDLX_ must be specified together; otherwise their values are computed.
- _ALPHA_ is optional but is recommended in order to maintain a complete set of decision limit information. The default value is 0.05.
- _LIMITK_ is optional. The default value is k, the number of groups. A group must have at least one nonmissing value ($n_i \geq 1$) and there must be at least one group with $n_i \geq 2$. If specified, _LIMITK_ overrides the value of k.
- _LIMITN_ is optional. The default value is the common group size (n), in the balanced case $n_i \equiv n$. If specified, _LIMITN_ overrides the value of n.
- The variable _TYPE_ is optional, but is recommended to maintain a complete set of decision limit information. The variable _TYPE_ must be a character variable of length 8. Valid values are `ESTIMATE`, `STANDARD`, `STDMEAN`, and `STDRMS`. The default is `ESTIMATE`.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- BY variables are required if specified with a BY statement.

SUMMARY= Data Set

You can read group summary statistics from a SUMMARY= data set specified in the PROC ANOM statement. This enables you to reuse OUTSUMMARY= data sets that have been created in previous runs of the ANOM procedure or to read output data sets created with SAS summarization procedures, such as PROC MEANS.

A SUMMARY= data set used with the BOXCHART statement must contain the following:

- the *group-variable*
- a group minimum variable for each *response*

- a group first-quartile variable for each *response*
- a group mean variable for each *response*
- a group median variable for each *response*
- a group third-quartile variable for each *response*
- a group maximum variable for each *response*
- a group standard deviation variable for each *response*
- a group sample size variable for each *response*

The names of the group summary statistics variables must be the *response* name concatenated with the following special suffix characters:

Group Summary Statistic	Suffix Character
group minimum	L
group first-quartile	1
group median	M
group mean	X
group third-quartile	3
group maximum	H
group standard deviation	S
group sample size	N

For example, consider the following statements:

```
proc anom summary=Summary;
   xchart (Weight Yldstren)*batch;
run;
```

The data set Summary must include the variables batch, WeightL, Weight1, WeightX, WeightM, Weight3, WeightH, WeightS, WeightN, YldsrenL, Yldsren1, YldsrenX, YldsrenM, Yldsren3, YldsrenH, YldsrenS, and YldsrenN. Note that if you specify a *response* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *response* name, suffixed with the appropriate character.

Other variables that can be read from a SUMMARY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the ANOM procedure reads all of the observations in a SUMMARY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option.

For an example of a SUMMARY= data set, see "Creating ANOM Boxcharts from Group Summary Data" on page 114.

TABLE= Data Set

You can read summary statistics and decision limits from a TABLE= data set specified in the PROC ANOM statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the ANOM procedure. Because the ANOM procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized ANOM charts.

The following table lists the variables required in a TABLE= data set used with the BOXCHART statement:

Table 5.21. Variables Required in a TABLE= Data Set

Variable	Description
group-variable	values of the *group-variable*
LDLX	lower decision limit for mean
LIMITN	nominal sample size associated with the decision limits
MEAN	central line
SUBMAX	group maximum
SUBMED	group median
SUBMIN	group minimum
SUBN	group sample size
SUBQ1	group first quartile
SUBQ3	group third quartile
SUBX	group mean
UDLX	upper decision limit for mean

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _VAR_. This variable is required if more than one *response* is specified or if the data set contains information for more than one *response*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Decision Limits" on page 118.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*group-variable*
Vertical	DATA=	*response*
Vertical	SUMMARY=	group mean variable
Vertical	TABLE=	_SUBX_

Missing Values

An observation read from a DATA=, SUMMARY=, or TABLE= data set is not analyzed if the value of the group variable is missing. For a particular response variable, an observation read from a DATA= data set is not analyzed if the value of the response variable is missing. Missing values of response variables generally lead to unequal group sample sizes. For a particular response variable, an observation read from a SUMMARY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples

This section provides an advanced example of the BOXCHART statement.

Example 5.1. ANOM Boxcharts with Unequal Group Sizes

See ANMBXEX1 in the SAS/QC Sample Library

Consider the example described in "Creating ANOM Boxcharts from Response Values" on page 112. Suppose that four of the 10 measurements were missing for the third and fourth labeler positions. The following statements create a SAS data set named LabelDev2, which contains the resulting deviation measurements:

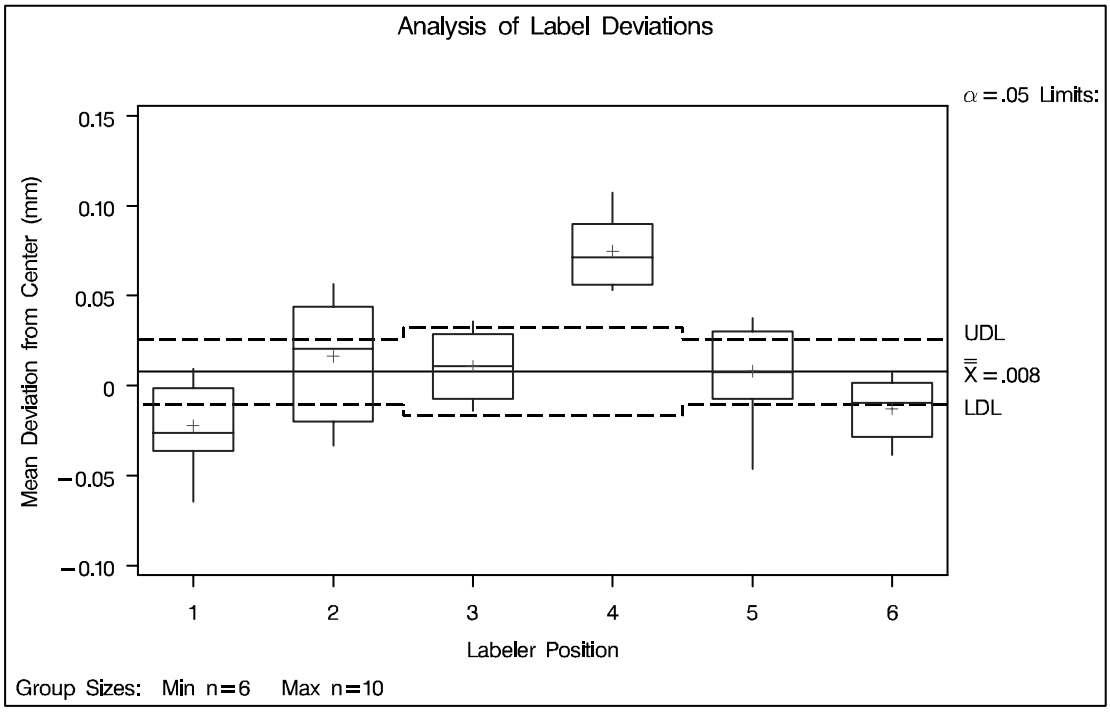
```
data LabelDev2;
   input Position @;
   do i = 1 to 5;
      input Deviation @;
      output;
   end;
   drop i;
   datalines;
1  -0.0239  -0.0285  -0.0300  -0.0043  -0.0362
1  -0.0422  -0.0014  -0.0647   0.0094  -0.0016
2  -0.0201  -0.0273   0.0227  -0.0332   0.0366
2   0.0438   0.0556   0.0098   0.0564   0.0182
3  -0.0073   0.0285    .        .      -0.0139
3    .       0.0357   0.0235   .      -0.0020
4   0.0669   0.1073    .        .       0.0755
4    .       0.0561   0.0899   .       0.0530
```

```
5    0.0368    0.0036    0.0374    0.0116   -0.0074
5    0.0250   -0.0080    0.0302   -0.0015   -0.0464
6    0.0049   -0.0384   -0.0204   -0.0049   -0.0120
6    0.0071   -0.0308    0.0017   -0.0285   -0.0070
run;
```

The following statements create the ANOM chart shown in Output 5.1.1:

```
title 'Analysis of Label Deviations';
proc anom data=LabelDev2;
   boxchart Deviation*Position;
   label Deviation = 'Mean Deviation from Center (mm)';
   label Position  = 'Labeler Position';
run;
```

Output 5.1.1. ANOM Chart with Unequal Group Sizes



Note that the decision limits are automatically adjusted for the varying group sizes. The legend reports the minimum and maximum group sizes.

Chapter 6
INSET Statement

Chapter Contents

OVERVIEW 143

GETTING STARTED 143
- Displaying Summary Statistics on an ANOM Chart 143
- Formatting Values and Customizing Labels 145
- Adding a Header and Positioning the Inset 146

SYNTAX 148
- Summary of INSET Keywords 149
- Summary of Options 149
- Dictionary of Options 150

DETAILS 152
- Positioning the Inset Using Compass Points 152
- Positioning the Inset in the Margins 153
- Positioning the Inset Using Coordinates 154

Chapter 6
INSET Statement

Overview

The INSET statement enables you to enhance an ANOM chart by adding a box or table (referred to as an *inset*) of summary statistics directly to the graph. An inset can display statistics calculated by the ANOM procedure or arbitrary values provided in a SAS data set.

Note that an INSET statement by itself does not produce a display but must be used in conjunction with a chart statement.

You can use options in the INSET statement to

- specify the position of the inset
- specify a header for the inset table
- specify graphical enhancements, such as background colors, text colors, text height, text font, and drop shadows

Getting Started

This section introduces the INSET statement with examples that illustrate commonly used options. Complete syntax for the INSET statement is presented in the "Syntax" section on page 148.

Displaying Summary Statistics on an ANOM Chart

A manufacturing engineer carries out a study to determine the source of excessive variation in the positioning of labels on shampoo bottles. * A labeling machine removes bottles from the line, attaches the labels, and returns the bottles to the line. There are six positions on the machine, and the engineer suspects that one or more of the position heads might be faulty.

See ANMIN1 in the SAS/QC Sample Library

A sample of 60 bottles, 10 per position, is run through the machine. For each bottle, the deviation of each label is measured in millimeters, and the machine position is recorded. The following statements create a SAS data set named LabelDeviations, which contains the deviation measurements for the 60 bottles:

*This example is based on a case study described by Hansen (1990).

```
data LabelDeviations;
   input Position @;
   do i = 1 to 5;
      input Deviation @;
      output;
   end;
   drop i;
   datalines;
1  -0.02386  -0.02853  -0.03001  -0.00428  -0.03623
1  -0.04222  -0.00144  -0.06466   0.00944  -0.00163
2  -0.02014  -0.02725   0.02268  -0.03323   0.03661
2   0.04378   0.05562   0.00977   0.05641   0.01816
3  -0.00728   0.02849  -0.04404  -0.02214  -0.01394
3   0.04855   0.03566   0.02345   0.01339  -0.00203
4   0.06694   0.10729   0.05974   0.06089   0.07551
4   0.03620   0.05614   0.08985   0.04175   0.05298
5   0.03677   0.00361   0.03736   0.01164  -0.00741
5   0.02495  -0.00803   0.03021  -0.00149  -0.04640
6   0.00493  -0.03839  -0.02037  -0.00487  -0.01202
6   0.00710  -0.03075   0.00167  -0.02845  -0.00697
;
run;
```

The following statements generate an ANOM chart from the LabelDeviations data. An INSET statement is used to display the mean square error (MSE) and the number of groups outside of the decision limits (NOUT) on the chart:

```
title 'ANOM Chart for Label Deviations';
proc anom data=LabelDeviations;
   xchart Deviation*Position;
   inset mse nout /
      height = 3;
run;
```

The resulting ANOM chart is displayed in Figure 6.1. The INSET statement immediately follows the chart statement that creates the graphical display (in this case, the XCHART statement). Specify the keywords for inset statistics (such as ALPHA) immediately after the word INSET. The inset statistics appear in the order in which you specify the keywords. The HEIGHT= option on the INSET statement specifies the text height used to display the statistics in the inset.

A complete list of keywords that you can use with the INSET statement is provided in "Summary of INSET Keywords" on page 149. Note that the set of keywords available for a particular display may depend on both the chart statement that precedes the INSET statement and the options that you specify in the chart statement.

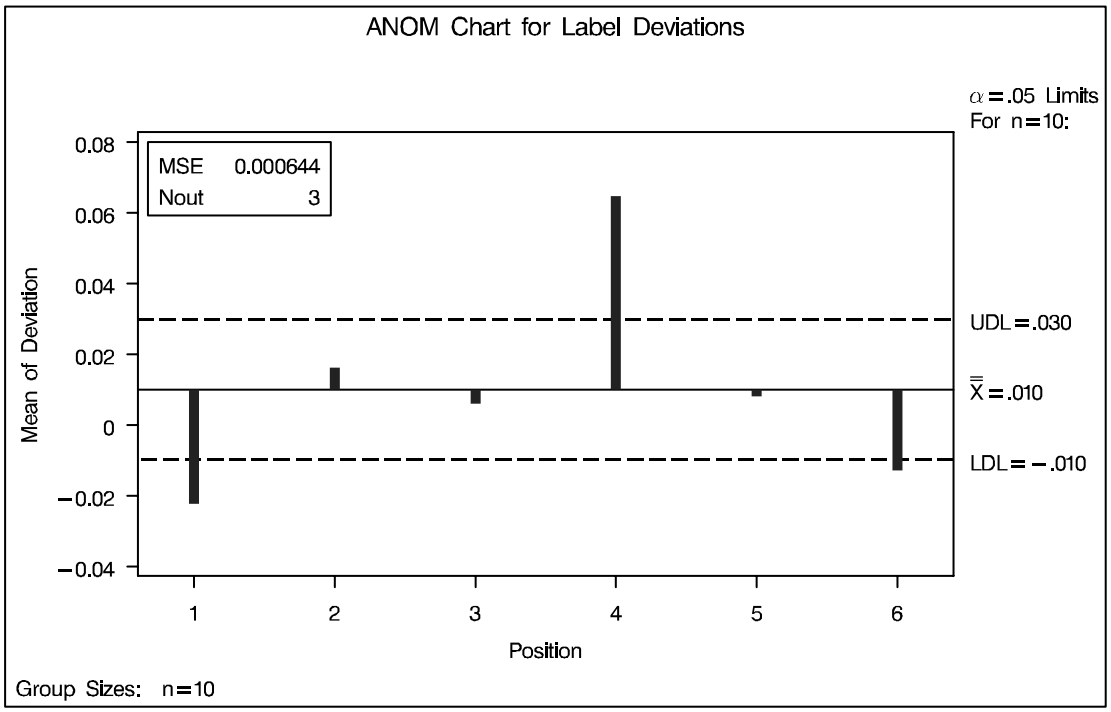


Figure 6.1. An ANOM Chart with an Inset

The following examples illustrate options commonly used for enhancing the appearance of an inset.

Formatting Values and Customizing Labels

See ANMIN2 in the SAS/QC Sample Library

By default, each inset statistic is identified with an appropriate label, and each numeric value is printed using an appropriate format. However, you may want to provide your own labels and formats. For example, in Figure 6.1 the default format used for the MSE prints an excessive number of decimal places. In the inset produced by the following statements, the unwanted decimal places are eliminated and the default MSE label is replaced by one specified by the user:

```
title 'ANOM Chart for Label Deviations';
proc anom data=LabelDeviations;
   xchart Deviation*Position;
   inset mse='Mean Square Error' (7.5) nout /
      height = 3;
run;
```

The resulting ANOM chart is displayed in Figure 6.2. You can provide your own label by specifying the keyword for that statistic followed by an equal sign (=) and the label in quotes. The label can have up to 24 characters.

The format 7.5 specified in parentheses after the MSE keyword displays the statistic with a field width of seven and five decimal places. In general, you can specify any

numeric SAS format in parentheses after an inset keyword. You can also specify a format to be used for all the statistics in the INSET statement with the FORMAT= option (see Figure 6.3). For more information about SAS formats, refer to Chapter 14 of *SAS Language Reference: Dictionary*.

Note that if you specify both a label and a format for a statistic, the label must appear before the format.

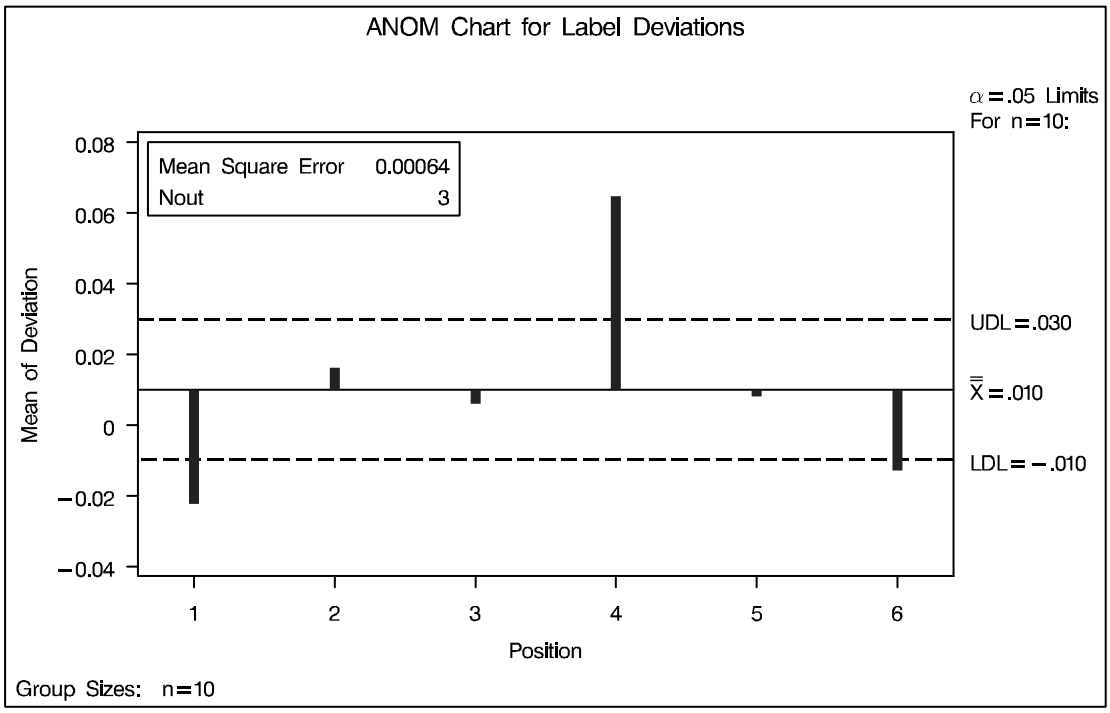


Figure 6.2. Formatting Values and Customizing Labels in an Inset

Adding a Header and Positioning the Inset

See ANMIN3 in the SAS/QC Sample Library

In the previous examples, the insets are displayed in the upper left corners of the plots, the default position for insets added to ANOM charts. You can control the inset position with the POSITION= option. In addition, you can display a header at the top of the inset with the HEADER= option. The following statements create a data set to be used with the INSET DATA= keyword and the chart shown in Figure 6.3:

```
data location;
   length _LABEL_ $ 10 _VALUE_ $ 12;
   input _LABEL_ _VALUE_ &;
   datalines;
Plant     Lexington
Line      1
Shift     2
;
```

```
title 'Mean Chart for Diameters';
proc anom data=LabelDeviations;
   xchart Deviation*Position;
   inset data=location mse nout /
      format   = 8.5
      position = rm
      cshadow  = black
      height   = 3
      header   = 'Summary';
run;
```

The header (in this case, *Summary*) can be up to 40 characters. POSITION=RM is specified to position the inset in the right margin. For more information about positioning, see "Details" on page 152. The CSHADOW= option is used to display a drop shadow on this inset. The *options*, such as HEADER=, POSITION=, and CSHADOW=, are specified after the slash (/) in the INSET statement. For more details on INSET statement options, see "Dictionary of Options" on page 150.

Note that the contents of the data set location appear before other statistics in the inset. The position of the DATA= keyword in the keyword list determines the position of the data set's contents in the inset. The FORMAT= option applies format 8.5 to the statistics listed in the INSET statement. Note that the format does not apply to the values from the location data set. You can associate a format with the _VALUE_ variable in the data set to format those values.

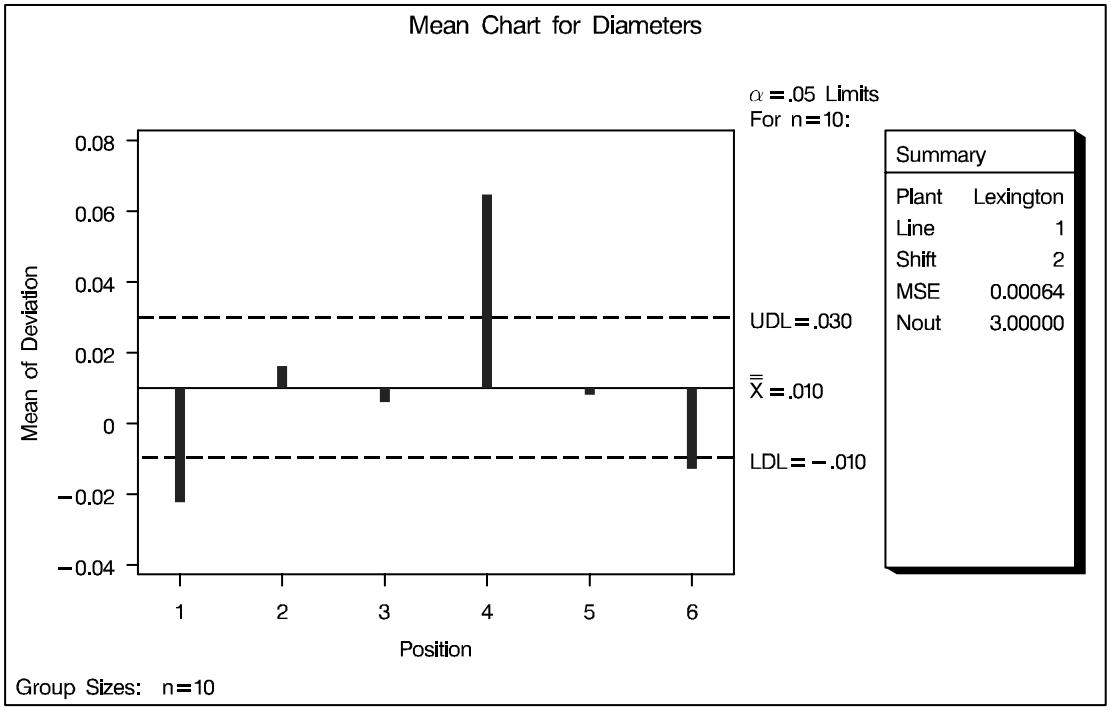


Figure 6.3. Adding a Header and Repositioning the Inset

Syntax

The syntax for the INSET statement is as follows:

INSET *keyword-list* < **/** *options* >**;**

You can use any number of INSET statements in the ANOM procedure. Each INSET statement produces a separate inset and must follow one of the chart statements. The inset appears on every panel (page) produced by the last chart statement preceding it. The statistics are displayed in the order in which they are specified. The following statements produce an ANOM boxchart with two insets and an ANOM chart for means with one inset.

```
proc anom data=LabelDeviations;
   boxchart Deviation*Position;
      inset alpha mse dfe;
      inset ldl mean udl;
   xchart Deviation*Position;
      inset ngroups nmin nmax;
run;
```

The statistics displayed in an inset are computed for a specific response variable using observations for the current BY group. For example, in the following statements, there are two response variables (weight and diameter) and a BY variable (location). If there are three different locations (levels of location), then a total of six ANOM charts are produced. The statistics in each inset are computed for a particular variable and location. The labels in the inset are the same for each ANOM chart.

```
proc anom data=axles;
   by location;
   xchart (weight diameter)*batch;
   inset alpha mse dfe;
run;
```

The components of the INSET statement are described as follows.

keyword-list
can include any of the *keywords* listed in "Summary of INSET Keywords" on page 149. By default, inset statistics are identified with appropriate labels, and numeric values are printed using appropriate formats. However, you can provide customized labels and formats. You provide the customized label by specifying the *keyword* for that statistic followed by an equal sign (=) and the label in quotes. Labels can have up to 24 characters. You provide the numeric format in parentheses after the *keyword*. Note that if you specify both a label and a format for a statistic, the label must appear before the format. For an example, see "Formatting Values and Customizing Labels" on page 145.

options

appear after the slash (/) and control the appearance of the inset. For example, the following INSET statement uses two appearance *options* (POSITION= and CTEXT=):

```
inset n nmin nmax / position=ne ctext=yellow;
```

The POSITION= option determines the location of the inset, and the CTEXT= option specifies the color of the text of the inset.

See "Summary of Options" on page 149 for a list of all available *options*, and "Dictionary of Options" on page 150 for detailed descriptions. Note the difference between *keywords* and *options*; *keywords* specify the information to be displayed in an inset, whereas *options* control the appearance of the inset.

Summary of INSET Keywords

All keywords available with the ANOM procedure's INSET statement request a single statistic in an inset, except for the DATA= keyword. The DATA= keyword specifies a SAS data set containing (label, value) pairs to be displayed in an inset. The data set must contain the variables _LABEL_ and _VALUE_. _LABEL_ is a character variable whose values provide labels for inset entries. _VALUE_ can be character or numeric, and provides values displayed in the inset. The label and value from each observation in the DATA= data set occupy one line in the inset. Figure 6.3 shows an inset containing entries from a DATA= data set.

Table 6.1. Summary Statistics

ALPHA	significance level
DATA=	(label, value) pairs from *SAS-data-set*
DFE	degrees of freedom
LDL	lower decision limit
MEAN	weighted average of group means
MSE	mean square error
N	nominal group size
NGROUPS	number of groups
NHIGH	number of groups above upper decision limit
NLOW	number of groups below lower decision limit
NMAX	maximum group size
NMIN	minimum group size
NOBS	total number of observations
NOUT	total number of groups outside decision limits
RMSE	root mean square error
UDL	upper decision limit

Summary of Options

The following table lists the INSET statement options. For complete descriptions, see "Dictionary of Options," which follows this section.

Table 6.2. INSET Options

CFILL=*color* | BLANK	specifies color of inset background
CFILLH=*color*	specifies color of header background
CFRAME=*color*	specifies color of frame
CHEADER=*color*	specifies color of header text
CSHADOW=*color*	specifies color of drop shadow
CTEXT=*color*	specifies color of inset text
DATA	specifies data units for POSITION=(x, y) coordinates
FONT=*font*	specifies font of text
FORMAT=*format*	specifies format of values in inset
HEADER=*'quoted string'*	specifies header text
HEIGHT=*value*	specifies height of inset text
NOFRAME	suppresses frame around inset
POSITION=*position*	specifies position of inset
REFPOINT=BR|BL|TR|TL	specifies reference point of inset positioned with POSITION=(x, y) coordinates

Dictionary of Options

The following entries provide detailed descriptions of options for the INSET statement. Terms used in this section are illustrated in Figure 6.4.

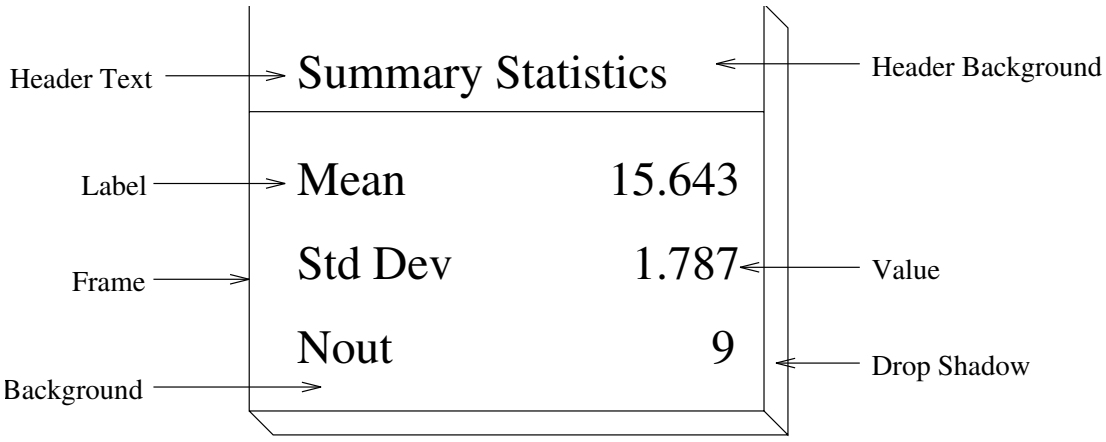


Figure 6.4. The Inset

CFILL=*color* | BLANK
specifies the color of the background (including the header background if you do not specify the CFILLH= option).

If you do not specify the CFILL= option, then by default, the background is empty. This means that items that overlap the inset (such as needles representing group

data or decision limits) show through the inset. If you specify any value for the CFILL= option, then overlapping items no longer show through the inset. Specify CFILL=BLANK to leave the background uncolored and also to prevent items from showing through the inset.

CFILLH=*color*
specifies the color of the header background. By default, if you do not specify a CFILLH= color, the CFILL= color is used.

CFRAME=*color*
specifies the color of the frame. By default, the frame is the same color as the axis of the plot.

CHEADER=*color*
specifies the color of the header text. By default, if you do not specify a CHEADER= color, the CTEXT= color is used.

CSHADOW=*color*
CS=*color*
specifies the color of the drop shadow. See Figure 6.3 on page 147 for an example. By default, if you do not specify the CSHADOW= option, a drop shadow is not displayed.

CTEXT=*color*
CT=*color*
specifies the color of the text. By default, the inset text color is the same as the other text on the plot.

DATA
specifies that data coordinates are to be used in positioning the inset with the POSITION= option. The DATA option is available only when you specify POSITION= (x, y), and it must be placed immediately after the coordinates (x, y). For details, see the entry for the POSITION= option or "Positioning the Inset Using Coordinates" on page 154. See Figure 6.7 on page 155 for an example.

FONT=*font*
specifies the font of the text. By default, the font is SIMPLEX if the inset is located in the interior of the plot, and the font is the same as the other text displayed on the plot if the inset is located in the exterior of the plot.

FORMAT=*format*
specifies a format for all the values displayed in an inset. If you specify a format for a particular statistic, then this format overrides the format you specified with the FORMAT= option.

HEADER= *'string'*
specifies the header text. The *string* cannot exceed 40 characters. If you do not specify the HEADER= option, no header line appears in the inset.

HEIGHT=*value*
specifies the height of the text.

NOFRAME
suppresses the frame drawn around the text.

POSITION=*position*
POS=*position*
determines the position of the inset. The *position* can be a compass point keyword, a margin keyword, or a pair of coordinates (x, y). You can specify coordinates in axis percent units or axis data units. For more information, see "Details" on page 152. By default, POSITION=NW, which positions the inset in the upper left (northwest) corner of the display.

REFPOINT=BR | BL | TR | TL
RP=BR | BL | TR | TL
specifies the reference point for an inset that is positioned by a pair of coordinates with the POSITION= option. Use the REFPOINT= option with POSITION= coordinates. The REFPOINT= option specifies which corner of the inset frame you want positioned at coordinates (x, y). The keywords BL, BR, TL, and TR represent bottom left, bottom right, top left, and top right, respectively. See Figure 6.8 on page 156 for an example. The default is REFPOINT=BL.

If you specify the position of the inset as a compass point or margin keyword, the REFPOINT= option is ignored. For more information, see "Positioning the Inset Using Coordinates" on page 154.

Details

This section provides details on three different methods of positioning the inset using the POSITION= option. With the POSITION= option, you can specify

- compass points
- keywords for margin positions
- coordinates in data units or percent axis units

Positioning the Inset Using Compass Points

See ANMIN4 in the SAS/QC Sample Library

You can specify the eight compass points N, NE, E, SE, S, SW, W, and NW as keywords for the POSITION= option. The following statements create the display in Figure 6.5, which demonstrates all eight compass positions. The default is NW.

```
title 'Mean Chart for Diameters';
proc anom data=LabelDeviations;
   xchart Deviation*Position;
   inset n / height=3 cfill=blank header='NW' pos=nw;
   inset n / height=3 cfill=blank header='N ' pos=n ;
   inset n / height=3 cfill=blank header='NE' pos=ne;
   inset n / height=3 cfill=blank header='E ' pos=e ;
   inset n / height=3 cfill=blank header='SE' pos=se;
   inset n / height=3 cfill=blank header='S ' pos=s ;
   inset n / height=3 cfill=blank header='SW' pos=sw;
   inset n / height=3 cfill=blank header='W ' pos=w ;
run;
```

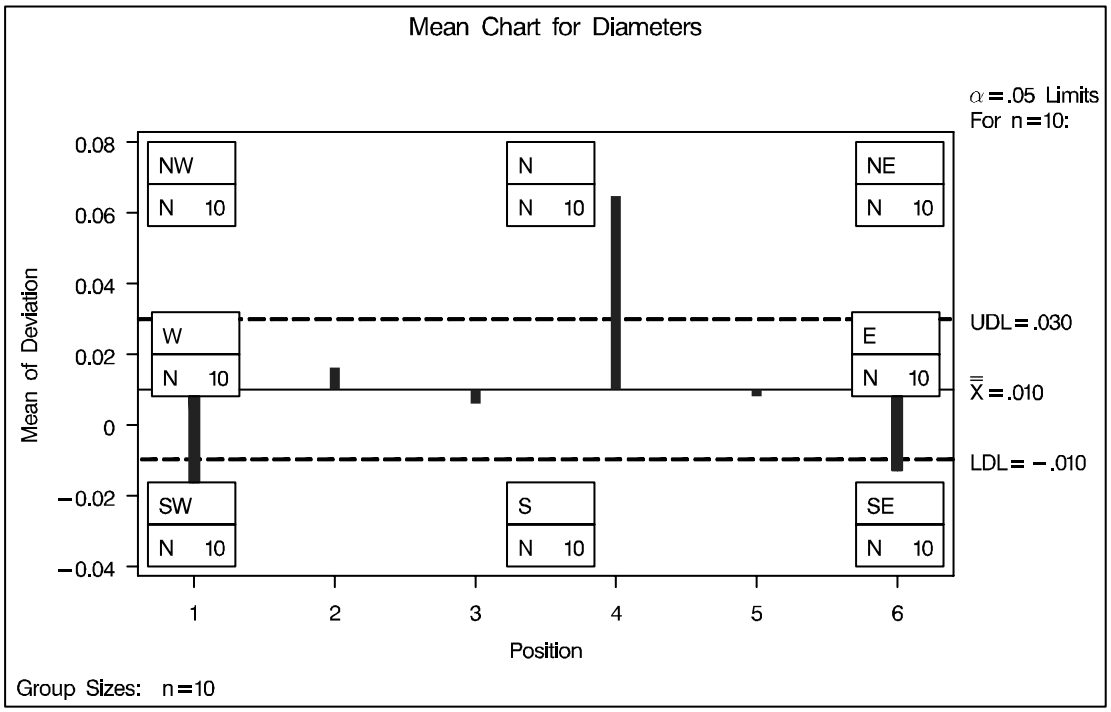


Figure 6.5. Insets Positioned Using Compass Points

Positioning the Inset in the Margins

Using the INSET statement you can also position an inset in one of the four margins surrounding the plot area using the margin keywords LM, RM, TM, or BM, as illustrated in Figure 6.6.

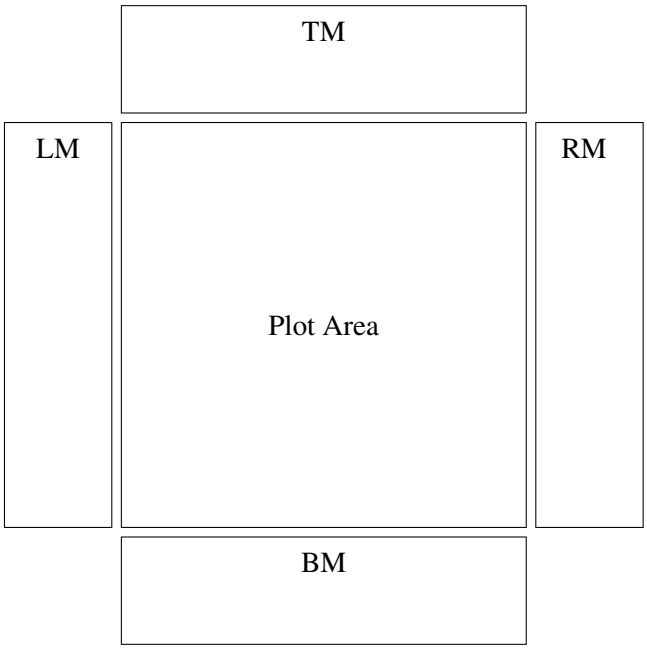


Figure 6.6. Positioning Insets in the Margins

For an example of an inset placed in the right margin, see Figure 6.3 on page 147. Margin positions are recommended if a large number of statistics are listed in the INSET statement. If you attempt to display a lengthy inset in the interior of the plot, it is likely that the inset will collide with the data display.

Positioning the Inset Using Coordinates

See ANMIN5 in the SAS/QC Sample Library

You can also specify the position of the inset with coordinates: POSITION= (x, y). The coordinates can be given in axis percent units (the default) or in axis data units.

Data Unit Coordinates

If you specify the DATA option immediately following the coordinates, the inset is positioned using axis data units. For example, the following statements place the bottom left corner of the inset at 2 on the horizontal axis and 0.04 on the vertical axis:

```
title 'Mean Chart for Diameters';
proc anom data=LabelDeviations;
   xchart Deviation*Position;
   inset n /
      header   = 'Position=(2,0.04)'
      height   = 3
      position = (2,0.04) data;
run;
```

The ANOM chart is displayed in Figure 6.7. By default, the specified coordinates determine the position of the bottom left corner of the inset. You can change this reference point with the REFPOINT= option, as in the next example.

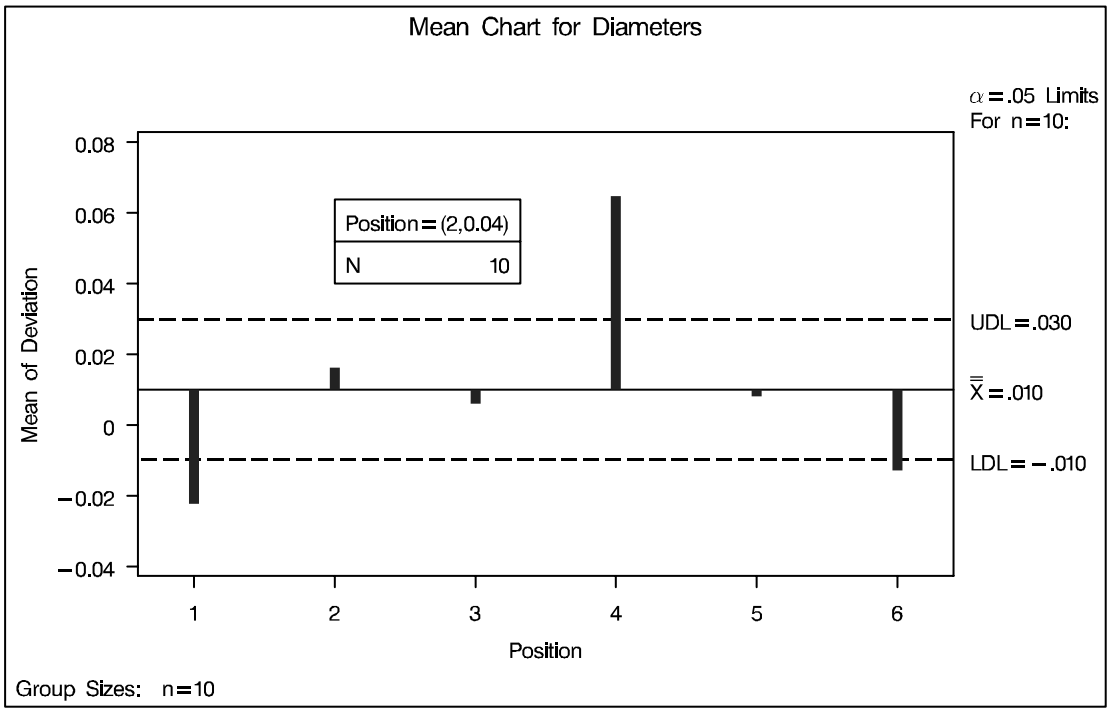


Figure 6.7. Inset Positioned Using Data Unit Coordinates

Axis Percent Unit Coordinates

If you do not use the DATA option, the inset is positioned using axis percent units. The coordinates of the bottom left corner of the display are $(0, 0)$, while the upper right corner is $(100, 100)$. For example, the following statements create an ANOM chart with two insets, both positioned using coordinates in axis percent units:

```
title 'Mean Chart for Diameters';
proc anom data=LabelDeviations;
   xchart Deviation*Position;
   inset mean / position = (5,25)
                header   = 'Position=(5,25)'
                height   = 3
                cfill    = blank
                refpoint = tl;
   inset mse / position = (95,95)
               header   = 'Position=(95,95)'
               height   = 3
               cfill    = blank
               refpoint = tr;
run;
```

The display is shown in Figure 6.8. Notice that the REFPOINT= option is used to determine which corner of the inset is to be placed at the coordinates specified with the POSITION= option. The first inset has REFPOINT=TL, so the top left corner of

the inset is positioned 5% of the way across the horizontal axis and 25% of the way up the vertical axis. The second inset has REFPOINT=TR, so the top right corner of the inset is positioned 95% of the way across the horizontal axis and 95% of the way up the vertical axis. Note also that coordinates in axis percent units must be *between* 0 and 100.

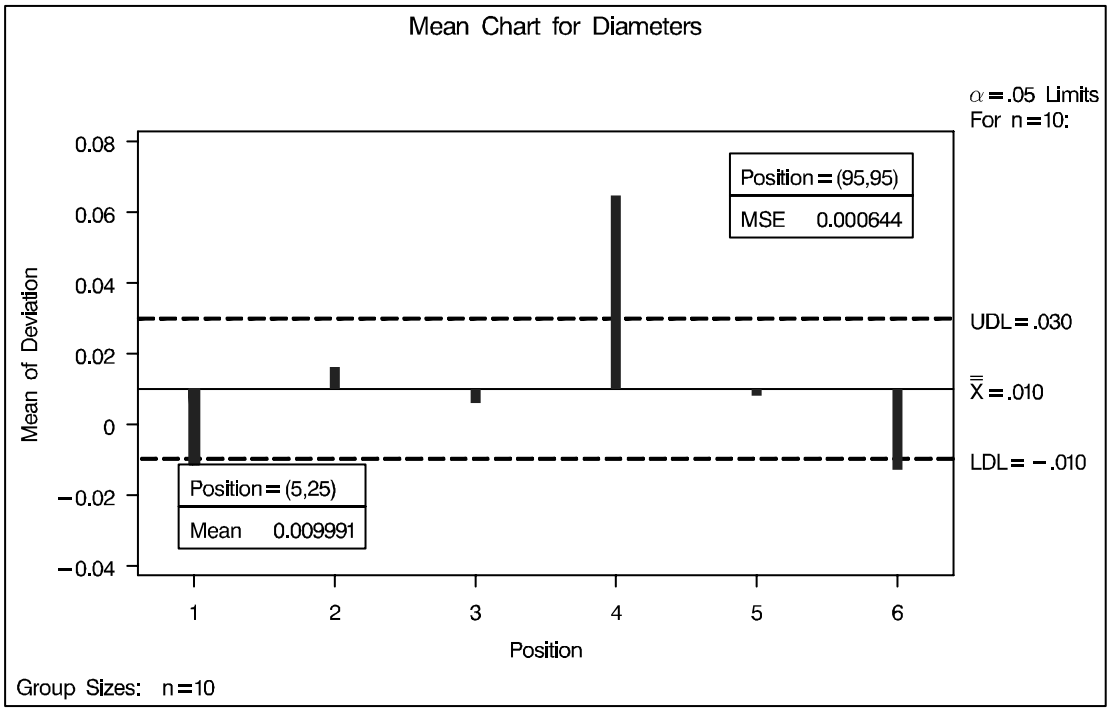


Figure 6.8. Inset Positioned Using Axis Percent Unit Coordinates

Chapter 7
References

Fritsch, K. and Hsu, J. (1997), "On Analysis of Means," *Advances in Statistical Decision Theory and Methodology*, Balakrishnan and Panchapakesan editors. Boston: Birkhauser.

Halperin, M., Greenhouse, S. W., Cornfield, J., and Zalokar, J. (1955), "Tables of Percentage Points for the Studentized Maximum Absolute Deviate in Normal Samples," *Journal of the American Statistical Association*, 50, 185–195.

Hansen, E. (1990), "Making the "Complex" Simple," *Problem-Driven Case Studies in Quality Improvement, Second Annual Symposium*, Madison, WI: Center for Quality and Productivity Improvement.

Laplace, P. S. (1827), "Mémoire sur le Flux et Reflux Lunaire Atmospherique," *Connaissance des Temps pour l'an,* 1830, 3–18.

Nelson, L. S. (1983), "Exact Critical Values for Use with the Analysis of Means," *Journal of Quality Technology*, 15, 40–44.

Nelson, P. R. (1981), "Numerical Evaluation of an Equicorrelated Multivariate Non-Central t Distribution," *Communications in Statistics, Part B –Simulation and Computation*, B10, 41–50.

Nelson, P. R. (1982a), "Exact Critical Points for the Analysis of Means," *Communications in Statistics*, A11, 699–709.

Nelson, P. R. (1982b), "Multivariate Normal and t Distributions with $\rho_{jk} = \alpha_j \alpha_k$," *Communications in Statistics–Simulation and Computation*, 11(2), 239–248.

Nelson, P. R. (1991), "Numerical Evaluation of Multivariate Normal Integrals with Correlations $\rho_{lj} = -\alpha_l \alpha_j$," *The Frontiers of Statistical Scientific Theory & Industrial Applications*, 97–114.

Nelson, P. R. (1993), "Additional Uses for the Analysis of Means and Extended Tables of Critical Values," *Technometrics*, 35, 61–71.

Nelson, P. R., Coffin, M., and Copeland, K. (2003), *Introductory Statistics for Engineering Experimentation,* Burlington, MA: Academic Press.

Ott, E. R. (1967), "Analysis of Means—A Graphical Procedure," *Industrial Quality Control*, 24, 101–109. Reprinted in *Journal of Quality Technology*, 15 (1983), 10–18.

Ott, E. R. (1975), *Process Quality Control: Troubleshooting and Interpretation of Data,* New York: McGraw-Hill.

Ramig, P. F. (1983), "Application of the Analysis of Means," *Journal of Quality Technology*, 15, 19–25.

Rodriguez, R. N. (1996), "Health Care Applications of Statistical Process Control: Examples Using the SAS System," *SAS Users Group International: Proceedings of the Twenty-First Annual Conference*, 1381–1396.

Soong, W. C. and Hsu, J. C. (1997), "Using Complex Integration to Compute Multivariate Normal Probabilities," *Journal of Computational and Graphical Statistics*, 6, 397–415.

In addition to the research literature listed above, the development of the ANOM procedure has benefited significantly from discussions with Jason Hsu (The Ohio State University) and Peter R. Nelson (Clemson University).

Part 2
The CAPABILITY Procedure

Contents

Introduction 161

Chapter 8. PROC CAPABILITY and General Statements 163

Chapter 9. CDFPLOT Statement 225

Chapter 10. COMPHISTOGRAM Statement 245

Chapter 11. HISTOGRAM Statement 277

Chapter 12. INSET Statement 353

Chapter 13. INTERVALS Statement 377

Chapter 14. OUTPUT Statement 391

Chapter 15. PPPLOT Statement 407

Chapter 16. PROBPLOT Statement 429

Chapter 17. QQPLOT Statement 461

References 501

Introduction

A process capability analysis compares the distribution of output from an in-control process to its specification limits to determine the consistency with which the specifications can be met. The CAPABILITY procedure provides the following:

- process capability indices, such as C_p and C_{pk}
- descriptive statistics based on moments, including skewness and kurtosis. Other descriptive information provided includes quantiles or percentiles (such as the median), frequency tables, and details on extreme values.
- histograms and comparative histograms. Optionally, these can be superimposed with specification limits, fitted probability density curves for various distributions, and kernel density estimates.
- cumulative distribution function plots (cdf plots). Optionally, these can be superimposed with specification limits and probability distribution curves for various distributions.
- quantile-quantile plots (Q-Q plots), probability plots, and probability-probability plots (P-P plots). These plots facilitate the comparison of a data distribution with various theoretical distributions. Optionally, Q-Q plots and probability plots can be superimposed with specification limits.
- goodness-of-fit tests for a variety of distributions including the normal. The assumption of normality is critical to the interpretation of capability indices.
- statistical intervals (prediction, tolerance, and confidence intervals) for a normal population
- the ability to produce plots either on a line printer or on graphics devices. Plots produced on graphics devices can be saved, replayed, and annotated.
- the ability to inset summary statistics and capability indices in plots produced on a graphics device
- the ability to analyze data sets with a frequency variable
- the ability to read specification limits from a data set
- the ability to create output data sets containing summary statistics, capability indices, histogram intervals, parameters of fitted curves, and statistical intervals

You can use the PROC CAPABILITY statement, together with the VAR and SPEC statements, to compute summary statistics and process capability indices. See "Getting Started" on page 166 for introductory examples. In addition, you can use the statements summarized in the following table to request plots and specialized analyses:

Statement	Result	Getting Started
CDFPLOT	cumulative distribution function plot	page 227
COMPHISTOGRAM	comparative histogram	page 247
HISTOGRAM	histogram	page 279
INSET	inset table on plot	page 355
INTERVALS	statistical intervals	page 379
OUTPUT	output data set with summary statistics and capability indices	page 393
PPPLOT	probability-probability plot	page 410
PROBPLOT	probability plot	page 431
QQPLOT	quantile-quantile plot	page 463

You can use the INSET statement with any of the plot statements to enhance the plot with an inset table of summary statistics. The INSET statement is applicable only to plots produced on graphics devices.

Learning about the CAPABILITY Procedure

To learn about the CAPABILITY procedure, first select the appropriate statement from the preceding table. Then turn to the corresponding "Getting Started" section for introductory examples.

To broaden your knowledge of the procedure, read Chapter 8, "PROC CAPABILITY and General Statements," which summarizes the syntax for the entire procedure and describes the PROC CAPABILITY statement, the VAR statement, and the SPEC statement. Subsequent chapters describe the statements listed in the preceding table. In addition to introductory examples, each chapter provides syntax summaries, descriptions of options, computational details, and advanced examples. Although the chapters are self-contained, much of what you learn about one plot statement, including the syntax, is transferable to other plot statements.

Chapter 8
PROC CAPABILITY and General Statements

Chapter Contents

OVERVIEW . . . 165

GETTING STARTED . . . 166
- Computing Descriptive Statistics . . . 166
- Computing Capability Indices . . . 169

SYNTAX OVERVIEW FOR THE CAPABILITY PROCEDURE . . . 170
- BY, FREQ, WEIGHT, and ID Statements . . . 171
- Graphical Enhancement Statements . . . 173

SYNTAX FOR THE PROC CAPABILITY STATEMENT . . . 173
- Summary of Options . . . 173
- Dictionary of Options . . . 176

SYNTAX FOR THE SPEC STATEMENT . . . 183
- Summary of Options . . . 184
- Dictionary of Options . . . 185

DETAILS . . . 187
- Input Data Sets . . . 187
- Output Data Set . . . 190
- Descriptive Statistics . . . 192
- Signed Rank Statistic . . . 194
- Tests for Normality . . . 194
- Percentile Computations . . . 197
- Robust Estimators . . . 200
- Computing the Mode . . . 203
- Assumptions and Terminology for Capability Indices . . . 203
- Standard Capability Indices . . . 204
- Specialized Capability Indices . . . 208
- Missing Values . . . 215
- ODS Tables . . . 216

EXAMPLES . . . 218
- Example 8.1. Reading Specification Limits . . . 218
- Example 8.2. Enhancing Reference Lines . . . 220
- Example 8.3. Displaying a Confidence Interval for Cpk . . . 222

Chapter 8
PROC CAPABILITY and General Statements

Overview

This chapter describes several statements that are generally used with the CAPABILITY procedure:

- The PROC CAPABILITY statement is required to invoke the CAPABILITY procedure. You can use this statement by itself to compute summary statistics.
- The VAR statement, which is optional, specifies the variables in the input data set that are to be analyzed. By default, all of the numeric variables are analyzed.
- The SPEC statement, which is optional, provides specification limits for the variables that are to be analyzed. When you use a SPEC statement, the procedure computes process capability indices in addition to summary statistics. Furthermore, the specification limits are displayed in plots created with plot statements (such as HISTOGRAM) that are described in subsequent chapters.

You can use the PROC CAPABILITY statement to request a variety of statistics for summarizing the data distribution of each analysis variable:

- sample moments
- basic measures of location and variability
- confidence intervals for the mean, standard deviation, and variance
- tests for location
- tests for normality
- trimmed and Winsorized means
- robust estimates of scale
- quantiles and related confidence intervals
- extreme observations and extreme values
- frequency counts for observations
- missing values

You can use the PROC CAPABILITY and SPEC statements together to request a variety of statistics for process capability analysis:

- percents of measurements within and outside specification limits
- confidence intervals for the probabilities of exceeding the specification limits
- standard capability indices and related confidence intervals
- tests of normality in conjunction with capability indices
- specialized capability indices

In addition, you can use options in the PROC CAPABILITY statement to

- specify the input data set to be analyzed
- specify an input data set containing specification limits
- specify a graphics catalog for saving graphical output
- specify rounding units for variable values
- specify the definition used to calculate percentiles
- specify the divisor used to calculate variances and standard deviations
- request that plots be produced on line printers and define special printing characters used for features
- suppress tables

You can use options in the SPEC statement to

- provide lower and upper specification limits and target values
- control the appearance of specification lines on plots
- control the appearance of the areas under a histogram outside the specification limits

Getting Started

This section introduces the PROC CAPABILITY, VAR, and SPEC statements with examples that illustrate the most commonly used options.

Computing Descriptive Statistics

See CAPPROC in the SAS/QC Sample Library

The fluid weights of 100 drink cans are measured in ounces. The filling process is assumed to be in statistical control. The measurements are saved in a SAS data set named CANS.

```
data cans;
   label weight = "Fluid Weight (ounces)";
   input WEIGHT @@;
datalines;
12.07  12.02  12.00  12.01  11.98  11.96  12.04  12.05  12.01  11.97
12.03  12.03  12.00  12.04  11.96  12.02  12.06  12.00  12.02  11.91
12.05  11.98  11.91  12.01  12.06  12.02  12.05  11.90  12.07  11.98
12.02  12.11  12.00  11.99  11.95  11.98  12.05  12.00  12.10  12.04
12.06  12.04  11.99  12.06  11.99  12.07  11.96  11.97  12.00  11.97
12.09  11.99  11.95  11.99  11.99  11.96  11.94  12.03  12.09  12.03
11.99  12.00  12.05  12.04  12.05  12.01  11.97  11.93  12.00  11.97
12.13  12.07  12.00  11.96  11.99  11.97  12.05  11.94  11.99  12.02
11.95  11.99  11.91  12.06  12.03  12.06  12.05  12.04  12.03  11.98
12.05  12.05  12.11  11.96  12.00  11.96  11.96  12.00  12.01  11.98
;
run;
```

You can use the PROC CAPABILITY and VAR statements to compute summary statistics for the weights.

```
title 'Process Capability Analysis of Fluid Weight';
proc capability data=cans normaltest;
   var weight;
run;
```

The input data set is specified with the DATA= option. The NORMALTEST option requests tests for normality. The VAR statement specifies the variables to analyze. If you omit the VAR statement, all numeric variables in the input data set are analyzed.

The descriptive statistics* for WEIGHT are shown in Figure 8.1. For instance, the average weight (labeled *Mean*) is 12.0093. The Shapiro-Wilk test statistic labeled *W* is 0.987876, and the probability of a more extreme value of W (labeled *Pr* < *W*) is 0.499. Compared to the usual cutoff value of 0.05, this probability (referred to as a *p*-value) indicates that the weights are normally distributed.

*In Version 7, the *Moments table has been reorganized. Tables for Basic Statistical Measures, Tests for Location, and Tests for Normality have been added.*

Process Capability Analysis of Fluid Weight

The CAPABILITY Procedure
Variable: weight (Fluid Weight (ounces))

Moments

N	100	Sum Weights	100
Mean	12.0093	Sum Observations	1200.93
Std Deviation	0.04695269	Variance	0.00220456
Skewness	0.05928405	Kurtosis	-0.1717404
Uncorrected SS	14422.5469	Corrected SS	0.218251
Coeff Variation	0.39096946	Std Error Mean	0.00469527

Basic Statistical Measures

Location		Variability	
Mean	12.00930	Std Deviation	0.04695
Median	12.00000	Variance	0.00220
Mode	12.00000	Range	0.23000
		Interquartile Range	0.07000

Tests for Location: Mu0=0

Test	-Statistic-		-----p Value------	
Student's t	t	2557.745	Pr > |t|	<.0001
Sign	M	50	Pr >= |M|	<.0001
Signed Rank	S	2525	Pr >= |S|	<.0001

Tests for Normality

Test	--Statistic---		-----p Value-----	
Shapiro-Wilk	W	0.987876	Pr < W	0.499
Kolmogorov-Smirnov	D	0.088506	Pr > D	0.052
Cramer-von Mises	W-Sq	0.079055	Pr > W-Sq	0.218
Anderson-Darling	A-Sq	0.457672	Pr > A-Sq	>0.250

Quantiles (Definition 5)

Quantile	Estimate
100% Max	12.130
99%	12.120
95%	12.090
90%	12.065
75% Q3	12.050
50% Median	12.000
25% Q1	11.980
10%	11.955
5%	11.935
1%	11.905
0% Min	11.900

Extreme Observations

-----Lowest----		----Highest----	
Value	Obs	Value	Obs
11.90	28	12.09	59
11.91	83	12.10	39
11.91	23	12.11	32
11.91	20	12.11	93
11.93	68	12.13	71

Figure 8.1. Descriptive Statistics

Computing Capability Indices

This example is a continuation of the previous example and shows how you can provide specification limits with a SPEC statement to request capability indices in addition to descriptive statistics.

See CAPPROC in the SAS/QC Sample Library

```
proc capability data=cans normaltest freq;
   spec lsl=11.95 target=12 usl=12.05;
   var weight;
run;
```

The options LSL=, TARGET=, and USL= specify the lower specification limit, target value, and upper specification limit for the weights. These statements produce the output shown in Figure 8.2 in addition to the output shown in Figure 8.1.

The CAPABILITY Procedure
Variable: weight (Fluid Weight (ounces))

Specification Limits

--------Limit--------		------Percent-------	
Lower (LSL)	11.95000	% < LSL	7.00000
Target	12.00000	% Between	77.00000
Upper (USL)	12.05000	% > USL	16.00000

Process Capability Indices

Index	Value	95% Confidence Limits	
Cp	0.354967	0.305565	0.404288
CPL	0.420991	0.332644	0.508117
CPU	0.288943	0.211699	0.365112
Cpk	0.288943	0.212210	0.365677
Cpm	0.348203	0.301472	0.398228

Frequency Counts

		Percents				Percents	
Value	Count	Cell	Cum	Value	Count	Cell	Cum
11.90	1	1.0	1.0	12.02	6	6.0	62.0
11.91	3	3.0	4.0	12.03	6	6.0	68.0
11.93	1	1.0	5.0	12.04	6	6.0	74.0
11.94	2	2.0	7.0	12.05	10	10.0	84.0
11.95	3	3.0	10.0	12.06	6	6.0	90.0
11.96	8	8.0	18.0	12.07	4	4.0	94.0
11.97	6	6.0	24.0	12.09	2	2.0	96.0
11.98	6	6.0	30.0	12.10	1	1.0	97.0
11.99	10	10.0	40.0	12.11	2	2.0	99.0
12.00	11	11.0	51.0	12.13	1	1.0	100.0
12.01	5	5.0	56.0				

Figure 8.2. Capability Indices and Frequency Table

In Figure 8.2, the table labeled *Specification Limits* lists the specification limits and target value, together with the percents of observations outside and between the lim-

its. The table labeled *Process Capability Indices* lists estimates for the standard process capability indices* C_p, CPL, CPU, C_{pk}, and C_{pm}, along with 95% confidence limits. The index C_{pm} is not computed unless you specify a TARGET= value. See page 204 for formulas used to compute the indices.

If you specify more than one variable in the VAR statement, you can provide corresponding specification limits and target values by specifying lists of values for the LSL=, USL=, and TARGET= options. As an alternative to the SPEC statement, you can read specification limits and target values from a data set specified with the SPEC= option in the PROC CAPABILITY statement. This is illustrated in Example 8.1 on page 218.

The FREQ option in the PROC CAPABILITY statement requests the table labeled *Frequency Counts* in Figure 8.2.

Syntax Overview for the CAPABILITY Procedure

The following are the primary statements that control the CAPABILITY procedure:

PROC CAPABILITY < *options* >;

VAR *variables*;

SPEC <*options* >;

CDFPLOT <*variables* > < **/** *options* >;

COMPHISTOGRAM <*variables* > **/ CLASS=(***class-variables***)** <*options* >;

HISTOGRAM <*variables* > < **/** *options* >;

PPPLOT <*variables* > < **/** *options* >;

PROBPLOT <*variables* > < **/** *options* >;

QQPLOT <*variables* > < **/** *options* >;

INSET *keyword-list* < **/** *options* >;

*In Release 6.12 and earlier releases, the capability index k was also included in this table; this index is now provided in the table labeled *Specialized Capability Indices*, which you can request with the SPECIALINDICES option.

INTERVALS <*variables* > < **/** *options* >**;**

OUTPUT <**OUT=***SAS-data-set*> *keyword=names* < . . . *keyword=names* >**;**

The PROC CAPABILITY statement invokes the procedure. The VAR statement specifies the numeric variables to be analyzed, and it is required if the OUTPUT statement is used to save summary statistics and capability indices in an output data set. If you do not use the VAR statement, all numeric variables in the data set are analyzed. The SPEC statement provides specification limits.

The plot statements (CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PPPLOT, PROBPLOT, and QQPLOT) create graphical displays, and the INSET statement enhances these displays by adding a table of summary statistics directly on the graph. The INTERVALS statement computes statistical intervals. You can specify one or more of each of the plot statements, the INSET statement, the INTERVALS statement, and the OUTPUT statement. If you use a VAR statement, the variables listed in a plot statement must be a subset of the variables listed in the VAR statement.

BY, FREQ, WEIGHT, and ID Statements

In addition, you can optionally specify one of each of the following statements:

BY *variables*;

FREQ *variable*;

WEIGHT *variable*;

ID *variable*;

The BY statement specifies variables in the input data set that are used for BY processing. A separate analysis is done for each group of observations defined by the levels of the BY variables. The input data set must be sorted in order of the BY variables.

The FREQ statement names a variable that provides frequencies for each observation in the input data set. If n is the value of the FREQ variable for a given observation, then that observation is used n times. If the value of the FREQ variable is missing or is less than one, the observation is not used in the analysis. If the value is not an integer, only the integer portion is used.

The WEIGHT statement names a variable that provides weights for each observation in the input data set. The CAPABILITY procedure uses the values w_i of the WEIGHT variable to modify the computation of a number of summary statistics by assuming that the variance of the ith value X_i of the analysis variable is equal to σ^2/w_i, where σ is an unknown parameter. This assumption is rarely applicable in process capability analysis, and the purpose of the WEIGHT statement is simply to make the

CAPABILITY procedure consistent with other data summarization procedures, such as the UNIVARIATE procedure.

The values of the WEIGHT variable do not have to be integers and are typically positive. By default, observations with non-positive or missing values of the WEIGHT variable are handled as follows*:

- If the value is zero, the observation is counted in the total number of observations.
- If the value is negative, it is converted to zero, and the observation is counted in the total number of observations.
- If the value is missing, the observation is excluded from the analysis.

To exclude observations that contain negative and zero weights from the analysis, specify the option EXCLNPWGT in the PROC statement. Note that most SAS/STAT procedures, such as PROC GLM, exclude negative and zero weights by default.

When you specify a WEIGHT variable, the procedure uses its values, w_i, to compute weighted versions of the statistics† provided in the *Moments* table. For example, the procedure computes a weighted mean $\overline{X}_w$ and a weighted variance s_w^2 as $\overline{X}_w = \frac{\sum_i w_i x_i}{\sum_i w_i}$ and $s_w^2 = \frac{1}{d}\sum_i w_i(x_i - \overline{X}_w)^2$ where x_i is the *i*th variable value. The divisor d is controlled by the VARDEF= option in the PROC CAPABILITY statement.

When you use both the WEIGHT and SPEC statements, capability indices are computed using $\overline{X}_w$ and s_w in place of $\overline{X}$ and s. Again, note that weighted capability indices are seldom needed in practice.

When you specify a WEIGHT statement, the procedure also computes a weighted standard error and a weighted version of Student's t test. This test is the only test of location that is provided when weights are specified.

The WEIGHT statement does not affect the determination of the mode, extreme values, extreme observations, or the number of missing values of the analysis variables. However, the weights w_i are used to compute weighted percentiles‡.

The WEIGHT variable has no effect on the calculation of extreme values, and it has no effect on graphical displays produced with the plot statements.

*In Release 6.12 and earlier releases, observations were used in the analysis if and only if the WEIGHT variable value was greater than zero.

†In Release 6.12 and earlier releases, weighted skewness and kurtosis were not computed.

‡In Release 6.12 and earlier releases, the weights did not affect the computation of percentiles.

Graphical Enhancement Statements

You can use TITLE, FOOTNOTE, and NOTE statements to enhance printed output. If you are creating plots with a graphics device, you can also use AXIS, LEGEND, PATTERN, and SYMBOL statements to enhance your plots. For details, see SAS/GRAPH documentation and the chapter for the plot statement that you are using.

Syntax for the PROC CAPABILITY Statement

The syntax for the PROC CAPABILITY statement is as follows:

PROC CAPABILITY < *options* >;

The following section lists all *options*. See "Dictionary of Options" on page 176 for detailed information.

Summary of Options

The following tables list all the PROC CAPABILITY *options* by function.

Table 8.1. Input Data Set Options

ANNOTATE=*SAS-data-set*	specifies input data set containing annotation information
DATA=*SAS-data-set*	specifies input data set
EXCLNPWGT	specifies that non-positive weights are to be excluded
NOBYSPECS	specifies that specification limits in SPEC= data set are to be applied to all BY groups
SPEC=*SAS-data-set*	specifies input data set with specification limits

Table 8.2. Plotting and Graphics Options

FORMCHAR(*index*)='*string*'	defines characters used for features on plots
GOUT=*graphics-catalog*	specifies catalog for saving graphical output
LINEPRINTER	requests line printer plots

Table 8.3. Computational Options

PCTLDEF=n	specifies definition used to calculate percentiles
ROUND=*round-units*	specifies units used to round variable values
VARDEF=*keyword*	specifies divisor used to calculate variances and standard deviations

Table 8.4. Data Summary Options

ALL	requests all tables
FREQ	requests frequency table
MODES	requests table of modes
NEXTROBS=*n*	requests table of n lowest, n highest observations
NEXTRVAL=*n*	requests table of n lowest, n highest values

Table 8.5. Output Options

NOPRINT	suppresses printed output
OUTTABLE=*SAS-data-set*	creates an output data set containing univariate statistics and capability indices in tabular form

Table 8.6. Hypothesis Testing Options

MU0=*value*	specifies mean for null hypothesis in tests for location
LOCCOUNT	requests table of counts used in sign test and signed rank test
NORMALTEST	performs tests for normality

Table 8.7. Robust Estimation Options

ROBUSTSCALE	requests table of robust measures of scale
TRIMMED=(*trimmed -options*)	requests table of trimmed means
WINSORIZED=(*Winsorized -options*)	requests table of Winsorized means

Table 8.8. TRIMMED-Options

ALPHA=*value*	specifies confidence level
TYPE=LOWER|UPPER| TWOSIDED	specifies type of confidence limit

Table 8.9. WINSORIZED-Options

ALPHA=*value*	specifies confidence level
TYPE=LOWER|UPPER| TWOSIDED	specifies type of confidence limit

Table 8.10. Capability Index Options

CPMA=*a*	(obsolete) specifies a for Cpm(a)
CHECKINDICES (*checkindices-options*)	requests test of normality in conjunction with standard indices
SPECIALINDICES	requests table of specialized indices including Boyles' C_{pm}, C_{jkp}, C_{pmk}, $C_{pm}(a)$, and Wright's C_s

Table 8.11. CHECKINDICES-Options

TEST=SW|KS|AD|CVM| NONE	specifies test for normality (Shapiro-Wilk, Kolmogorov-Smirnov, Anderson-Darling, Cramér-von Mises, or no test)

Table 8.12. Confidence Limit Options

ALPHA=*value*	specifies level for all confidence limits
CIBASIC (*cibasic-options*)	requests confidence limits for the mean, standard deviation, variance
CIINDICES (*ciindices-options*)	specifies level and type of confidence limits for capability indices
CIPCTLDF (*cipctldf-options*)	requests distribution-free confidence limits for percentiles
CIPCTLNORMAL (*cipctlnormal-options*)	requests confidence limits for percentiles assuming normality
CIPROBEX (*ciprobex-options*)	requests confidence limits for the probability of exceeding specifications

Table 8.13. CIBASIC-Options

ALPHA=*value*	specifies confidence level
TYPE=LOWER|UPPER| TWOSIDED	specifies type of confidence limit

Table 8.14. CIINDICES-Options

ALPHA=*value*	specifies confidence level
TYPE=LOWER|UPPER| TWOSIDED	specifies type of confidence limit

Table 8.15. CIPCTLDF-Options

ALPHA=*value*	specifies confidence level
TYPE=LOWER|UPPER| SYMMETRIC| ASYMMETRIC	specifies type of confidence limit

Table 8.16. CIPCTLNORMAL-Options

ALPHA=*value*	specifies confidence level
TYPE=LOWER|UPPER| TWOSIDED	specifies type of confidence limit

Table 8.17. CIPROBEX-Options

ALPHA=*value*	specifies confidence level
TYPE=LOWER|UPPER| TWOSIDED	specifies type of confidence limit

Dictionary of Options

The following entries provide detailed descriptions of the *options* in the PROC CAPABILITY statement. The marginal notes *Graphics* and *Line Printer* identify options that apply to graphics devices and line printers, respectively.

ALL

requests all of the tables generated by the FREQ, MODES, NEXTRVAL=5, CIBASIC, CIPCTLDF, and CIPCTLNORMAL options. If a WEIGHT statement is not used, the ALL option also requests the tables generated by the LOCCOUNT, NORMALTEST, ROBUSTSCALE, TRIMMED=.25, and WINSORIZED=.25 options. PROC CAPABILITY uses any values that you specify with the ALPHA=, MUO=, NEXTRVAL=, CIBASIC, CIPCTLDF, CIPCTLNORMAL, TRIMMED=, or WINSORIZED= options in conjunction with the ALL option.

ALPHA=*value*

specifies the default confidence level for all confidence limits computed by the CAPABILITY procedure*. The coverage percent for the confidence limits is $(1 - value)100$. For example, ALPHA=0.10 results in 90% confidence limits. The default *value* is 0.05.

Note that specialized ALPHA= options are available for a number of confidence interval options. For example, you can specify CIBASIC(ALPHA=0.10) to request a table of *Basic Confidence Limits* at the 90% level. The default *values* of these options default to the value of the general ALPHA= option.

ANNOTATE=*SAS-data-set*
ANNO=*SAS-data-set*

Graphics

specifies an input data set containing annotate variables as described in SAS/GRAPH documentation. You can use this data set to add features to plots produced on graphics devices. Use this data set only when the chart is created using a graphics device; it is ignored when the LINEPRINTER option is specified. Features provided in this data set are added to every plot produced in the current run of the procedure.

CHECKINDICES<(<TEST = SW | KS | AD | CVM | NONE> <ALPHA=*value***>)>**

specifies the test of normality used in conjunction with process capability indices that are displayed in the *Process Capability Indices* table. The tests available are Shapiro-Wilk (SW), Kolmogorov-Smirnov (KS), Anderson-Darling (AD), and Cramér-von Mises (CVM). The default test is the Shapiro-Wilk test if the sample size is less than or equal to 2000 and the Kolmogorov-Smirnov test if the sample size is greater than 2000. If the *p*-value for the test is less than the cutoff probability *value* specified with the ALPHA= option†, a warning is added to the table, as illustrated in Figure 8.3. The *value* must be between zero and one, and typical values are 0.05 and 0.10. See "Tests for Normality" on page 194 for details concerning the test.

*In Release 6.12 and earlier release, the level for confidence limits for capability indices was specified with the GAMMA= option in the SPEC statement. This option still works but is considered obsolete.

†In Release 6.12 and earlier releases, this *value* was specified with the ALPHA= option in the SPEC statement. This option still works but is considered obsolete.

```
                          The CAPABILITY Procedure
                               Variable:  p2

                        Process Capability Indices

                  Index       Value     95% Confidence Limits

                  Cp       0.541072     0.388938     0.692946
                  CPL      0.642426     0.417087     0.862984
                  CPU      0.439718     0.257339     0.617184
                  Cpk      0.439718     0.259310     0.620126

Warning:  Normality is rejected for alpha = 0.05 using the Shapiro-Wilk test
```

Figure 8.3. Warning Message Printed with Capability Indices

CIBASIC<(<TYPE=*keyword***><ALPHA=***value***>)>**
requests confidence limits for the mean, standard deviation, and variance based on the assumption that the data are normally distributed. With large sample sizes, this assumption is not required for confidence limits for the mean.

TYPE=*keyword*
specifies the type of confidence limit, where *keyword* is LOWER, UPPER, or TWOSIDED. The default value is TWOSIDED.

ALPHA=*value*
specifies the confidence level. The coverage percent for the confidence limits is $(1 - value)100$. For example, ALPHA=0.10 requests 90% confidence limits. The default *value* is 0.05.

CIINDICES<(<TYPE=*keyword***><ALPHA=***value***>)>**
specifies the type and level of the confidence limits for standard capability indices displayed in the table labeled *Process Capability Indices*.

TYPE=*keyword*
specifies the type of confidence limit, where *keyword* is LOWER, UPPER, or TWOSIDED. The default value is TWOSIDED.

ALPHA=*value*
specifies the confidence level. The coverage percent for the confidence limits is $(1 - value)100$. For example, ALPHA=0.10 requests 90% confidence limits. The default value is 0.05.

CIPCTLDF<(TYPE=*keyword***><ALPHA=***value***>)>**
CIQUANTDF<(TYPE=*keyword***><ALPHA=***value***>)>**
requests confidence limits for quantiles using a distribution-free method. In other words, no specific parametric distribution (such as the normal) is assumed for the data. Order statistics are used to compute the confidence limits as described in Section 5.2 of Hahn and Meeker (1991). This option is not available if you specify a WEIGHT statement.

TYPE=*keyword*
specifies the type of confidence limit, where *keyword* is LOWER, UPPER, SYMMETRIC, or ASYMMETRIC. The default value is SYMMETRIC.

ALPHA=*value*
specifies the confidence level. The coverage percent for the confidence limits is $(1 - value)100$. For example, ALPHA=0.10 requests 90% confidence limits. The default *value* is 0.05.

CIPCTLNORMAL<(TYPE=*keyword***><ALPHA=***value***>)>**
CIQUANTNORMAL<(TYPE=*keyword***><ALPHA=***value***>)>**
requests confidence limits for quantiles based on the assumption that the data are normally distributed. The computational method is described in Section 4.4.1 of Hahn and Meeker (1991) and uses the noncentral t distribution as given by Odeh and Owen (1980). This option is not available if you specify a WEIGHT statement

CIPROBEX<(TYPE=*keyword***><ALPHA=***value***>)>**
requests confidence limits for $Pr[X \leq \text{LSL}]$ and $Pr[X \geq \text{USL}]$, where X is the analysis variable, LSL is the lower specification limit, and USL is the upper specification limit. The computational method, which assumes that X is normally distributed, is described in Section 4.5 of Hahn and Meeker (1991) and uses the noncentral t distribution as given by Odeh and Owen (1980). This option is not available if you specify a WEIGHT statement

TYPE=*keyword*
specifies the type of confidence limit, where *keyword* is LOWER, UPPER, or TWOSIDED. The default value is TWOSIDED.

ALPHA=*value*
specifies the confidence level. The coverage percent for the confidence limits is $(1 - value)100$. For example, ALPHA=0.10 requests 90% confidence limits. The default value is 0.05.

CPMA=*value*
specifies the *value* of the parameter a for the capability index $C_{pm}(a)$. This option has been superseded by the SPECIALINDICES(CPMA=) option.

DATA=*SAS-data-set*
specifies the input data set containing the observations to be analyzed. If the DATA= option is omitted, the procedure uses the most recently created SAS data set.

DEF=*index*
is an alias for the PCTLDEF= option. See the entry for the PCTLDEF= option.

EXCLNPWGT
excludes observations with non-positive weight values (zero or nonnegative) for the analysis. By default, PROC CAPABILITY treats observations with negative weights like those with zero weights and counts them in the total number of observations. This option is applicable only if you specify a WEIGHT statement.

FORMCHAR(*index***)=**'*string*'

defines characters used for features on plots, where *index* is a number ranging from 1 to 11, and *string* is a character or hexadecimal string. The *index* identifies which features are controlled with the *string* characters, as discussed in the table that follows. If you specify the FORMCHAR= option omitting the *index*, the *string* controls all 11 features. *Line Printer*

By default, the form character list specified with the SAS system option FORMCHAR= is used; otherwise, the default is FORMCHAR='|—-|+|—'. If you print to a PC screen or your device supports the ASCII symbol set (1 or 2), the following is recommended:

```
formchar='B3,C4,DA,C2,BF,C3,C5,B4,C0,C1,D9'X
```

As an example, suppose you want to plot the data values of the empirical cumulative distribution function with asterisks (*). You can change the appropriate character using the following:

```
formchar(2)='*'
```

Note that the FORMCHAR= option in the PROC CAPABILITY statement allows you to temporarily override the values of the SAS system option with the same name. The values of the SAS system option are not altered by using the FORMCHAR= option in PROC CAPABILITY statement.

The features associated with values of *index* are as follows:

Value of *index*	Description of Character	Chart Feature
1	vertical bar	frame, ecdf line, HREF= lines
2	horizontal bar	frame, ecdf line, VREF= lines
3	box character (upper left)	frame, ecdf line, histogram bars
4	box character (upper middle)	histogram bars, tick marks (horizontal axis)
5	box character (upper right)	frame, histogram bars
6	box character (middle left)	histogram bars
7	box character (middle middle)	not used
8	box character (middle right)	histogram bars, tick marks (vertical axis)
9	box character (lower left)	frame
10	box character (lower middle)	histogram bars
11	box character (lower right)	frame, ecdf line

FREQ
requests a frequency table in the printed output that contains the variable values, frequencies, percentages, and cumulative percentages. See Figure 8.2 on page 169 for an example.

GOUT=*graphics-catalog*
Graphics
specifies a graphics catalog in which to save graphics output.

LINEPRINTER
Line Printer
requests that line printer charts be produced. By default, the procedure creates charts for a graphics device.

LOCCOUNT
requests a table with the number of observations greater than, not equal to, and less than the value of MUO=. PROC CAPABILITY uses these values to construct the sign test and signed rank test. This option is not available if you specify a WEIGHT statement.

MODES
MODE
requests a table of all possible modes. By default, when the data contains multiple modes, PROC CAPABILITY displays the lowest mode in the table of basic statistical measures. When all values are unique, PROC CAPABILITY does not produce a table of modes.

MUO=*value(s)*
LOCATION=*value(s)*
specifies the *value* of the mean or location parameter (μ_o) in the null hypothesis for the tests summarized in the table labeled *Tests for Location: Mu0=value*. If you specify a single *value*, PROC CAPABILITY tests the same null hypothesis for all analysis variables. If you specify multiple *values*, a VAR statement is required, and PROC CAPABILITY tests a different null hypothesis for each analysis variable by matching the VAR variables with the *values* in the corresponding order. The default *value* is 0.

NEXTROBS=*n*
specifies the number of extreme observations in the table labeled *Extreme Observations*. The table lists the *n* lowest observations and the *n* highest observations. The default value is 5. The value of *n* must be an integer between 0 and half the number of observations. You can specify NEXTROBS=0 to suppress the table.

NEXTRVAL=*n*
requests the table labeled *Extreme Values* and specifies the number of extreme values in the table. The table lists the *n* lowest unique values and the *n* highest unique values. The value of *n* must be an integer between 0 and half the maximum number of observations. By default, $n = 0$ and no table is displayed.

NOPRINT
suppresses the tables of descriptive statistics and capability indices which are created by the PROC CAPABILITY statement. The NOPRINT option does not suppress the

tables created by the INTERVALS or plot statements. You can use the NOPRINT options in these statements to suppress the creation of their tables.

NORMALTEST
NORMAL
requests a table of *Tests for Normality* for each of the analysis variables. The table provides test statistics and *p*-values for the Shapiro-Wilk test (provided the sample size is less than or equal to 2000), the Kolmogorov-Smirnov test, the Anderson-Darling test, and the Cramér-von Mises test. See "Tests for Normality" on page 194 for details. If specification limits are provided, the NORMALTEST option is assumed.

OUTTABLE=*SAS-data-set*
specifies an output data set that contains univariate statistics and capability indices arranged in tabular form. See "OUTTABLE= Data Set" on page 190 for details.

PCTLDEF=*index*
DEF=*index*
specifies one of five definitions used to calculate percentiles. The value of *index* can be 1, 2, 3, 4, or 5. See "Percentile Computations" on page 197 for details. By default, PCTLDEF=5.

ROBUSTSCALE
requests a table of robust measures of scale. These measures include the interquartile range, Gini's mean difference, the median absolute deviation about the median (*MAD*), and two statistics proposed by Rousseeuw and Croux (1993), Q_n, and S_n. This option is not available if you specify a WEIGHT statement.

ROUND=*value-list*
specifies units used to round variable values. The ROUND= option reduces the number of unique values for each variable and hence reduces the memory required for temporary storage. *Values* must be greater than 0 for rounding to occur.

If you use only one *value*, the procedure uses this unit for all variables. If you use a list of values, you must also use a VAR statement. The procedure then uses the roundoff values for variables in the order given in the VAR statement. For example, the following statements specify a roundoff value of 1 for YLDSTREN and a roundoff value of 0.5 for TENSTREN.

```
proc capability round=1 0.5;
   var yldstren tenstren;
run;
```

When a variable value is midway between the two nearest rounded points, the value is rounded to the nearest even multiple of the roundoff value. For example, with a roundoff value of 1, the variable values of -2.5, -2.2, and -1.5 are rounded to -2; the values of -0.5, 0.2, and 0.5 are rounded to 0; and the values of 0.6, 1.2, and 1.4 are rounded to 1.

SPECIALINDICES

requests a table of specialized process capability indices. These indices include k, Boyles' modified C_{pm} (also denoted as $C_{pm}+$), C_{jkp}, $C_{pm}(a)$, $C_p(5.15)$, $C_{pk}(5.15)$, C_{pmk}, Wright's C_s,

Boyles' S_{jkp}, C_{pp}, C''_{pp}, C_{pg}, C_{pq}, C_p^W, C_{pk}^W, C_{pm}^W, C_{pc}, and Vännmann's $C_p(u,v)$ and $C_p(v)$.

You can provide values for the parameters a for $C_{pm}(a)$, u and v for $C_p(u,v)$ and $C_p(v)$, and for the γ multiplier for C_s by specifying the following options in parentheses after the SPECIALINDICES option.

CPMA=*value*

specifies the *value* of the parameter a for the capability index $C_{pm}(a)$ described in Section 3.7 of Kotz and Johnson (1993). The *value* must be positive. The default *value* is 0.5. The existing CPMA= option in the PROC CAPABILITY statement is considered obsolete but still works.

CPU=*value*

specifies the *value* of the parameter u for Vännmann's capability index $C_p(u,v)$. The *value* must be greater than or equal to zero. The default *value* is zero.

CPV=*value*

specifies the *value* of the parameter v for Vännmann's capability indices $C_p(u,v)$ and $C_p(v)$. The *value* must be greater than or equal to zero. The default *value* is 4.

CSGAMMA=*value*

specifies the *value* of the γ multiplier suggested by Chen and Kotz (1996) for Wright's capability index C_s. The *value* must be greater than zero. The default *value* is 1.

SPECS=*SAS-data-set*
SPEC=*SAS-data-set*

specifies an input data set containing specification limits for each of the variables in the VAR statement. This option is an alternative to the SPEC statement, which also provides specification limits. See "SPEC= Data Set" on page 188 for details on SPEC= data sets, and Example 8.1 on page 218 for an example. If you use both the SPEC= option and a SPEC statement, the SPEC= option is ignored.

TRIMMED=*values(s)* **<(TYPE=***keyword***><ALPHA=***value***>)>**
TRIM=*values(s)* **<(TYPE=***keyword***><ALPHA=***value***>)>**

requests a table of trimmed means, where each *value* specifies the number or the proportion of trimmed observations. If the *value* is the number n of trimmed observations, n must be between 0 and half the number of nonmissing observations. If the *value* is a proportion *p* between 0 and 0.5, the number of observations trimmed is the smallest integer greater than or equal to *np*, where *n* is the number of observations. To obtain confidence limits for the mean and the student *t*-test, you must use the default value of VARDEF= which is DF. The TRIMMED= option is not available if you specify a WEIGHT statement.

TYPE=*keyword*
specifies the type of confidence limit, where *keyword* is LOWER, UPPER, or TWOSIDED. The default value is TWOSIDED.

ALPHA=*value*
specifies the confidence level. The coverage percent is $(1 - value)100$. For example, ALPHA=0.10 requests a 90% confidence limit. The default value is 0.05.

VARDEF=DF | N | WDF | WEIGHT | WGT
specifies the divisor used in calculating variances and standard deviations. The values and associated divisors are shown in the following table. By default, VARDEF=DF.

Value	Divisor	Formula
DF	degrees of freedom	$n - 1$
N	number of observations	n
WEIGHT | WGT	sum of weight	$\sum_i w_i$
WDF	sum of weights minus one	$(\sum_i w_i) - 1$

WINSORIZED=*values(s)* **<(TYPE=***keyword***><ALPHA=***value***>)>**
WINSOR=*values(s)* **<(TYPE=***keyword***><ALPHA=***value***>)>**
requests a table of winsorized means, where each *value* specifies the number or the proportion of winsorized observations. If the *value* is the number n of winsorized observations, n must be between 0 and half the number of nonmissing observations. If the *value* is a proportion p between 0 and 0.5, the number of observations winsorized is the smallest integer greater than or equal to np, where n is the number of observations. To obtain confidence limits for the mean and the student t-test, you must use the default value of VARDEF= which is DF. The WINSORIZED= option is not available if you specify a WEIGHT statement.

TYPE=*keyword*
specifies the type of confidence limit, where *keyword* is LOWER, UPPER, or TWOSIDED. The default value is TWOSIDED.

ALPHA=*value*
specifies the confidence level. The coverage percent is $(1 - value)100$. For example, ALPHA=0.10 results in a 90% confidence limit. The default value is 0.05.

Syntax for the SPEC Statement

The syntax for the SPEC statement is as follows:

SPEC < *options* >;

You can use at most one SPEC statement in the CAPABILITY procedure. When you provide specification limits and target values in a SPEC statement, the tabular output produced by the PROC CAPABILITY statement includes process capability indices as well as summary statistics. You can use the SPEC statement in conjunction with

the CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PROBPLOT, and QQPLOT statements to add specification limit and target lines to the plots produced with these statements.

options

control features of the specification limits and target values. The "Summary of Options" section, which follows, lists all options by function. The "Dictionary of Options" section on page 185 describes each option in more detail.

Summary of Options

The following tables list the SPEC statement *options* by function. For complete descriptions see the "Dictionary of Options" on page 185.

Table 8.18. Lower Specification Limit Options

CLEFT=*color*	color of pattern used to fill area left of lower specification limit (histograms only)
CLSL=*color*	color of lower specification limit line
LLSL=*linetype*	line type of lower specification limit line
LSL=*value-list*	lower specification limit values
LSLSYMBOL='*character*'	character used to plot lower specification limit line on line printers
PLEFT=*pattern*	pattern type used to fill area left of lower specification limit (histograms only)
WLSL=n	width of lower specification limit line

Table 8.19. Target Options

CTARGET=*color*	color of target line
LTARGET=*linetype*	line type of target line
TARGET=*value-list*	target value
TARGETSYMBOL='*character*'	character used to plot target on line printers
WTARGET=n	width of target line

Table 8.20. Upper Specification Limit Options

CRIGHT=*color*	color of pattern used to fill area right of upper specification limit (histograms only)
CUSL=*color*	color of upper specification limit line
LUSL=*linetype*	line type of upper specification limit line
PRIGHT=*pattern*	pattern type used to fill area right of upper specification limit (histograms only)
USL=*value-list*	upper specification limit values
USLSYMBOL='*character*'	character used to plot upper specification limit on line printers
WUSL=n	width of upper specification limit line

Dictionary of Options

The following entries provide detailed descriptions of options for the SPEC statement. The marginal notes *Graphics* and *Line Printer* identify options that can be used only with graphics devices and line printers, respectively.

CLEFT=*color*

specifies the color of the pattern used to fill the area under a histogram to the left of the lower specification limit. This option is applicable only when the SPEC statement is used in conjunction with a HISTOGRAM or COMPHISTOGRAM statement. See Output 8.2.1 on page 222 for an example. If a pattern is specified for this area with the PLEFT= option, the default color is the second color in the device color list. The CLEFT= option also applies to the area under a fitted curve;for an example, see Output 11.1.1 on page 335. *Graphics*

CLSL=*color*

specifies the color of the lower specification line displayed in plots created with the CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PROBPLOT, and QQPLOT statements. The default is the second color in the device color list. *Graphics*

CRIGHT=*color*

specifies the color of the pattern used to fill the area under a histogram to the right of the upper specification limit. This option is applicable only when the SPEC statement is used in conjunction with a HISTOGRAM or COMPHISTOGRAM statement. See Output 8.2.1 on page 222 for an example. If a pattern is specified for this area with the PRIGHT= option, the default color is the third color in the device color list. The CRIGHT= option also applies to the area under a fitted curve; for an example, see Output 11.1.1 on page 335. *Graphics*

CTARGET=*color*

specifies the color of the target line displayed in plots created with the CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PROBPLOT, and QQPLOT statements. The default is the first color in the device color list. *Graphics*

CUSL=*color*

specifies the color of the upper specification line displayed in plots created with the CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PROBPLOT, and QQPLOT statements. The default is the third color in the device color list. *Graphics*

LLSL=*linetype*

specifies the line type for the lower specification line displayed in plots created with the CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PROBPLOT, and QQPLOT statements. See Output 8.2.1 on page 222 for an example. The default is 1, which produces a solid line. *Graphics*

LSL=*value-list*

specifies the lower specification limits for the variables listed in the VAR statement, or for all numeric variables in the input data set if no VAR statement is used. If you specify only one lower limit, it is used for all of the variables; otherwise, the number of limits must match the number of variables. See "Computing Capability Indices" on page 169 for an example.

LSLSYMBOL='*character***'**

Line Printer

specifies the *character* used to display the lower specification line in line printer plots created with the CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PROBPLOT, and QQPLOT statements. The default character is 'L'.

LTARGET=*linetype*

Graphics

specifies the line type for the target line in plots created with the CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PROBPLOT, and QQPLOT statements. See Output 8.2.1 on page 222 for an example. The default is 1, which produces a solid line.

LUSL=*linetype*

Graphics

specifies the line type for the upper specification line displayed in plots created with the CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PROBPLOT, and QQPLOT statements. See Output 8.2.1 on page 222 for an example. The default is 1, which produces a solid line.

PLEFT=*pattern*

Graphics

specifies the pattern used to fill the area under a histogram to the left of the lower specification limit. This option is applicable only when the SPEC statement is used in conjunction with a HISTOGRAM or COMPHISTOGRAM statement. For an example, see Output 8.2.1 on page 222. The PLEFT= option also applies to the area under a fitted curve; for an example, see Output 11.1.1 on page 335. If a CLEFT= color is specified, the default pattern is a solid fill.

PRIGHT=*pattern*

Graphics

specifies the pattern used to fill the area under a histogram to the right of the upper specification limit. This option is applicable only when the SPEC statement is used in conjunction with a HISTOGRAM or COMPHISTOGRAM statement. For an example, see Output 8.2.1 on page 222. The PRIGHT= option also applies to the area under a fitted curve; for an example, see Output 11.1.1 on page 335. If a CRIGHT= color is specified, the default pattern is a solid fill.

TARGET=*value-list*

specifies a target values for the variables listed in the VAR statement, or for all numeric variables in the input data set if no VAR statement is used. If you specify only one target value, it is used for all of the variables; otherwise, the number of values must match the number of variables. See "Computing Capability Indices" on page 169 for an example.

TARGETSYMBOL='*character***'**
TARGETSYM='*character***'**

Line Printer

specifies the *character* used to display the target line in line printer plots created with the CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PROBPLOT, and QQPLOT statements. The default character is 'T'.

USL=*value-list*

specifies the upper specification limits for the variables listed in the VAR statement, or for all numeric variables in the input data set if no VAR statement is used. If you specify only one upper limit, it is used for all of the variables; otherwise, the number

of limits must match the number of variables. See "Computing Capability Indices" on page 169 for an example.

USLSYMBOL='*character***'**
specifies the *character* used to display the upper specification line in line printer plots created with the CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PROBPLOT, and QQPLOT statements. The default character is 'U'. *Line Printer*

WLSL=*n*
specifies the width in pixels of the lower specification line in plots created with the CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PROBPLOT, and QQPLOT statements. See Output 8.2.1 on page 222 for an illustration. The default is 1. *Graphics*

WTARGET=*n*
specifies the width in pixels of the target line in plots created with the CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PROBPLOT, and QQPLOT statements. See Output 8.2.1 on page 222 for an illustration. The default is 1. *Graphics*

WUSL=*n*
specifies the width in pixels of the upper specification line in plots created with the CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PROBPLOT, and QQPLOT statements. See Output 8.2.1 on page 222 for an illustration. The default is 1. *Graphics*

Details

This section provides details on the following topics:

- input data sets specified with the DATA= option, the SPEC= option, and the ANNOTATE= option
- the output data set specified with the OUTTABLE= option
- descriptive statistics
- the tests for normality requested with the NORMALTEST option
- percentile definitions controlled using the PCTLDEF= option
- robust estimators
- computing the mode
- assumptions and terminology for capability indices
- standard capability indices
- specialized capability indices

Input Data Sets

DATA= Data Set

The DATA= data set contains a set of variables that represent measurements from a process. The CAPABILITY procedure must have a DATA= data set. If you do not specify one with the DATA= option in the PROC CAPABILITY statement, the procedure uses the last data set created.

SPEC= Data Set

The SPEC= option in the PROC CAPABILITY statement identifies a SPEC= data set, which contains specification limits. This option is an alternative to using the SPEC statement. If you use both the SPEC= option and a SPEC statement, the SPEC= option is ignored. The SPEC= option is especially useful when:

- the number of variables is large
- the same specification limits are referred to in more than one analysis
- a BY statement is used
- batch processing is used

The following variables are read from a SPEC= data set:

Variable	Description
LSL	lower specification limit
TARGET	target value
USL	upper specification limit
VAR	name of the variable

You may omit either _LSL_ or _USL_ but not both. _TARGET_ is optional. If the SPEC= data set contains both _LSL_ and _USL_, you can assign missing values to _LSL_ or _USL_ to indicate one-sided specifications. You can assign missing values to _TARGET_ when the variable does not use a target value. _LSL_, _USL_, and _TARGET_ must be numeric variables. _VAR_ must be a character variable.

You can include the following optional variables in a SPEC= data set to control the appearance of specification limits on charts:

Variable	Description
CLEFT	color of pattern used to fill area left of LSL (histograms only)
CLSL	color of LSL line
CRIGHT	color of pattern used to fill area right of USL (histograms only)
CTARGET	color of target line
CUSL	color of USL line
LLSL	line type of LSL line
LSLSYM	character used to plot LSL line on line printers
LTARGET	line type of target line
LUSL	line type of USL line
PLEFT	pattern type used to fill area left of LSL (histograms only)
PRIGHT	pattern type used to fill area right of USL (histograms only)
TARGETSYM	character used to plot target on line printers
USLSYM	character used to plot USL on line printers
WLSL	width of LSL line
WTARGET	width of target line
WUSL	width of USL line

If you are using the HISTOGRAM statement to create "clickable" histograms in HTML, you can also provide the following variables in a SPEC= data set:

Variable	Description
LOURL	URL associated with area to left of lower specification limit
HIURL	URL associated with area to right of upper specification limit
URL	URL associated with area between specification limits

These are character variables whose values are Uniform Resource Locators (URLs) linked to areas on a histogram. When you view the ODS HTML output with a browser, you can click on an area, and the browser will bring up the page specified by the corresponding URL.

If you use a BY statement, the SPEC= data set must also contain the BY variables. The SPEC= data set must be sorted in the same order as the DATA= data set. Within a BY group, specification limits for each variable plotted are read from the first observation where _VAR_ matches the variable name.

See the "Examples" section on page 218 for an example of reading specification limits from a SPEC= data set.

ANNOTATE= Data Sets

In the CAPABILITY procedure, you can add features to plots by specifying ANNOTATE= data sets either in the PROC CAPABILITY statement or in individual plot statements. Depending on where you specify an ANNOTATE= data set, however, the information is used for all plots or only for plots produced by a given statement.

Information contained in the ANNOTATE= data set specified in the PROC CAPABILITY statement is used for all plots produced in a given PROC step; this is a "global" ANNOTATE= data set. By using this global data set, you can keep information common to all high-resolution plots in one data set.

Information contained in the ANNOTATE= data set specified in a plot statement is used for plots produced by that statement; this is a "local" ANNOTATE= data set. By using this data set, you can add statement-specific features to plots. For example, you can add different features to plots produced by the HISTOGRAM and QQPLOT statements by specifying an ANNOTATE= data set in each plot statement.

In addition, you can specify an ANNOTATE= data set in the PROC CAPABILITY statement and in plot statements. This allows you to add some features to all plots (those given in the data set specified in the PROC statement) and also add statement-specific features to plots (those given in the data set specified in the plot statement).

For complete details on the structure and content of Annotate type data sets, see SAS/GRAPH documentation.

Output Data Set

OUTTABLE= Data Set

The OUTTABLE= data set saves univariate statistics and capability indices. The following variables can be saved:

Variable	Description
_CP_	capability index C_p
CPK	capability index C_{pk}
CPL	capability index CPL
CPM	capability index C_{pm}
CPU	capability index CPU
_K_	capability index K
KURT	kurtosis
LSL	lower specification limit
MAX	maximum
MEAN	mean
MEDIAN	median
MIN	minimum
MODE	mode
NMISS	number of missing observations
_N_	number of nonmissing observations
_P1_	1^{st} percentile
_P5_	5^{th} percentile
_P10_	10^{th} percentile
_P90_	90^{th} percentile
_P95_	95^{th} percentile
_P99_	99^{th} percentile
PCTGTR	percentage of observations greater than upper specification limit
PCTLSS	percentage of observations less than lower specification limit
_Q1_	25^{th} percentile (lower quartile)
_Q3_	75^{th} percentile (upper quartile)
QRANGE	interquartile range (upper quartile minus lower quartile)
RANGE	range
SGNRNK	centered sign rank
SKEW	skewness
STD	standard deviation
SUMWGT	sum of the weights
SUM	sum
TARGET	target value
USL	upper specification limit
VARI	variance
VAR	variable name

Note: The variables _CP_, _CPK_, _CPL_, _CPM_, _CPU_, _K_, _LSL_, _PCTGTR_, _PCTLSS_, _TARGET_, and _USL_ are included if you provide specification limits.

The OUTTABLE= data set and the OUT= data set* starting on page 391 for details on the OUT= data set. contain essentially the same information. However, the structure of the OUTTABLE= data set may be more appropriate when you are computing summary statistics or capability indices for more than one process variable in the same invocation of the CAPABILITY procedure. Each observation in the OUTTABLE= data set corresponds to a different process variable, and the variables in the data set correspond to summary statistics and indices.

For example, suppose you have ten process variables (P1-P10). The following statements create an OUTTABLE= data set named TABLE, which contains summary statistics and capability indices for each of these variables:

See CAPTAB1 in the SAS/QC Sample Library

```
proc capability data=process outtable=table noprint;
  var p1-p10;
  specs lsl=5 10 65 35 35 5 25 25 60 15
        usl=175 275 300 450 550 200 275 425 500 525;
run;
```

The following statements create the table shown in Figure 8.4, which contains the mean, standard deviation, lower and upper specification limits, and capability index C_{pk} for each process variable:

```
proc print data=table label noobs;
   var _VAR_ _MEAN_ _STD_ _LSL_ _USL_ _CPK_;
   label _VAR_='Process';
run;
```

Process	Mean	Standard Deviation	Lower Specification Limit	Upper Specification Limit	Capability Index CPK
p1	90.76	57.024	5	175	0.49242
p2	167.32	81.628	10	275	0.43972
p3	224.56	96.525	65	300	0.26052
p4	258.08	145.218	35	450	0.44053
p5	283.48	157.033	35	550	0.52745
p6	107.48	52.437	5	200	0.58814
p7	153.20	90.031	25	275	0.45096
p8	217.08	130.031	25	425	0.49239
p9	280.68	140.943	60	500	0.51870
p10	243.24	178.799	15	525	0.42551

Figure 8.4. Tabulating Results for Multiple Process Variables

*See Chapter 14, "OUTPUT Statement,"

Descriptive Statistics

This section provides computational details for the descriptive statistics which are computed with the PROC CAPABILITY statement. These statistics can also be saved in the OUT= data set by specifying the keywords listed in Table 14.1 on page 399 in the OUTPUT statement.

Standard algorithms (Fisher 1973) are used to compute the moment statistics. The computational methods used by the CAPABILITY procedure are consistent with those used by other SAS procedures for calculating descriptive statistics. For details on statistics also calculated by Base SAS software, see *SAS Language Reference: Dictionary*.

The following sections give specific details on several statistics calculated by the CAPABILITY procedure.

Mean

The sample mean is calculated as

$$\frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$$

where n is the number of nonmissing values for a variable, x_i is the i^{th} value of the variable, and w_i is the weight associated with the i^{th} value of the variable. If there is no WEIGHT= variable, the formula reduces to $\frac{1}{n}\sum_{i=1}^{n} x_i$.

Sum

The sum is calculated as $\sum_{i=1}^{n} w_i x_i$, where n is the number of nonmissing values for a variable, x_i is the i^{th} value of the variable, and w_i is the weight associated with the i^{th} value of the variable. If there is no WEIGHT= variable, the formula reduces to $\sum_{i=1}^{n} x_i$.

Sum of the Weights

The sum of the weights is calculated as $\sum_{i=1}^{n} w_i$, where n is the number of nonmissing values for a variable and w_i is the weight associated with the i^{th} value of the variable. If there is no WEIGHT= variable, the sum of the weights is n.

Variance

The variance is calculated as

$$\frac{1}{d}\sum_{i=1}^{n} w_i (x_i - \bar{X}_w)^2$$

where n is the number of nonmissing values for a variable, x_i is the i^{th} value of the variable, $\bar{X}_w$ is the weighted mean, w_i is the weight associated with the i^{th} value of

the variable, and d is the divisor controlled by the VARDEF= option in the PROC CAPABILITY statement. If there is no WEIGHT= variable, the formula reduces to

$$\frac{1}{d}\sum_{i=1}^{n}(x_i - \bar{X}_w)^2$$

Standard Deviation

The standard deviation is calculated as

$$\sqrt{\frac{1}{d}\sum_{i=1}^{n} w_i(x_i - \bar{X}_w)^2}$$

where n is the number of nonmissing values for a variable, x_i is the i^{th} value of the variable, $\bar{X}_w$ is the weighted mean, w_i is the weight associated with the i^{th} value of the variable, and d is the divisor controlled by the VARDEF= option in the PROC CAPABILITY statement. If there is no WEIGHT= variable, the formula reduces to

$$\sqrt{\frac{1}{d}\sum_{i=1}^{n}(x_i - \bar{X}_w)^2}$$

Skewness

The sample skewness is calculated as

$$\frac{n}{(n-1)(n-2)}\sum_{i=1}^{n}\left(\frac{x_i - \bar{X}}{s}\right)^3$$

where n is the number of nonmissing values for a variable and must be greater than 2, x_i is the i^{th} value of the variable, $\bar{X}$ is the sample average, and s is the sample standard deviation.

The sample skewness can be positive or negative; it measures the asymmetry of the data distribution and estimates the theoretical skewness $\sqrt{\beta_1} = \mu_3{\mu_2}^{-\frac{3}{2}}$, where μ_2 and μ_3 are the second and third central moments. Observations that are normally distributed should have a skewness near zero.

Kurtosis

The sample kurtosis is calculated as

$$\frac{n(n+1)}{(n-1)(n-2)(n-3)}\sum_{i=1}^{n}\left(\frac{x_i - \bar{X}}{s}\right)^4 - \frac{3(n-1)^2}{(n-2)(n-3)}$$

where $n > 3$. The sample kurtosis measures the heaviness of the tails of the data distribution. It estimates the adjusted theoretical kurtosis denoted as $\beta_2 - 3$, where $\beta_2 = \frac{\mu_4}{{\mu_2}^2}$, and μ_4 is the fourth central moment. Observations that are normally distributed should have a kurtosis near zero.

Coefficient of Variation (CV)

The coefficient of variation is calculated as

$$CV = \frac{100 \times s}{\bar{X}}$$

Signed Rank Statistic

The signed rank statistic S is computed as

$$S = \sum_{i:x_i>0} r_i^+ - \frac{n(n+1)}{4}$$

where r_i^+ is the rank of $|x_i|$ after discarding values of $x_i = 0$, and n is the number of nonzero x_i values. Average ranks are used for tied values.

If $n \leq 20$, the significance of S is computed from the exact distribution of S, where the distribution is a convolution of scaled binomial distributions. When $n > 20$, the significance of S is computed by treating

$$S\sqrt{\frac{n-1}{nV - S^2}}$$

as a Student t variate with $n - 1$ degrees of freedom. V is computed as

$$V = \frac{1}{24}n(n+1)(2n+1) - \frac{1}{48}\sum t_i(t_i+1)(t_i-1)$$

where the sum is over groups tied in absolute value and where t_i is the number of values in the i^{th} group (Iman 1974, Conover 1980). The null hypothesis tested is that the mean (or median) is zero, assuming that the distribution is symmetric. Refer to Lehmann (1975).

Tests for Normality

You can use the NORMALTEST option in the PROC CAPABILITY statement to request several tests of the hypothesis that the analysis variable values are a random sample from a normal distribution. These tests, which are summarized in the table labeled *Tests for Normality*, include the following:

- Shapiro-Wilk test
- Kolmogorov-Smirnov test
- Anderson-Darling test
- Cramér-von Mises test

Tests for normality are particularly important in process capability analysis because the commonly used capability indices are difficult to interpret unless the data are at least approximately normally distributed. Furthermore, the confidence limits for capability indices displayed in the table labeled *Process Capability Indices* require the assumption of normality. Consequently, the tests of normality are always computed when you specify the SPEC statement, and a note is added to the table when the hypothesis of normality is rejected. You can specify the particular test and the significance level with the CHECKINDICES option.

Shapiro-Wilk Test

If the sample size is 2000 or less, * the procedure computes the Shapiro-Wilk statistic W (also denoted as W_n to emphasize its dependence on the sample size n). The statistic W_n is the ratio of the best estimator of the variance (based on the square of a linear combination of the order statistics) to the usual corrected sum of squares estimator of the variance. When n is greater than three, the coefficients to compute the linear combination of the order statistics are approximated by the method of Royston (1992). The statistic W_n is always greater than zero and less than or equal to one $(0 < W \leq 1)$.

Small values of W lead to rejection of the null hypothesis. The method for computing the p-value (the probability of obtaining a W statistic less than or equal to the observed value) depends on n. For $n = 3$, the probability distribution of W is known and is used to determine the p-value. For $n > 4$, a normalizing transformation is computed:

$$Z_n = \begin{cases} (-\log(\gamma - \log(1 - W_n)) - \mu)/\sigma & \text{if } 4 \leq n \leq 11 \\ (\log(1 - W_n) - \mu)/\sigma & \text{if } 12 \leq n \leq 2000 \end{cases}$$

The values of σ, γ, and μ are functions of n obtained from simulation results. Large values of Z_n indicate departure from normality, and since the statistic Z_n has an approximately standard normal distribution, this distribution is used to determine the p-values for $n > 4$.

EDF Tests for Normality

The Kolmogorov-Smirnov, Anderson-Darling and Cramér-von Mises tests for normality are based on the empirical distribution function (EDF) and are often referred to as EDF tests. EDF tests for a variety of non-normal distributions are available in the HISTOGRAM statement; see the "EDF Goodness-of-Fit Tests" section on page 323 for details. For a thorough discussion of these tests, refer to D'Agostino and Stephens (1986).

The empirical distribution function is defined for a set of n independent observations $X_1, \ldots, X_n$ with a common distribution function $F(x)$. Under the null hypothesis,

*In Release 6.12 and earlier releases, the CAPABILITY procedure performed a Shapiro-Wilk test for sample sizes of 2000 or smaller, and a Kolmogorov-Smirnov test otherwise. The computed value of W was used to interpolate linearly within the range of simulated critical values given in Shapiro and Wilk (1965). In Version 7, minor improvements have been made to the algorithm for the Shapiro-Wilk test, as described in this section.

$F(x)$ is the normal distribution. Denote the observations ordered from smallest to largest as $X_{(1)}, \ldots, X_{(n)}$. The empirical distribution function, $F_n(x)$, is defined as

$$F_n(x) = \begin{cases} 0, & x < X_{(1)} \\ \frac{i}{n}, & X_{(i)} \leq x < X_{(i+1)}, i = 1, \ldots, n-1 \\ 1, & X_{(n)} \leq x \end{cases}$$

Note that $F_n(x)$ is a step function that takes a step of height $\frac{1}{n}$ at each observation. This function estimates the distribution function $F(x)$. At any value x, $F_n(x)$ is the proportion of observations less than or equal to x, while $F(x)$ is the probability of an observation less than or equal to x. EDF statistics measure the discrepancy between $F_n(x)$ and $F(x)$.

The EDF tests make use of the probability integral transformation $U = F(X)$. If $F(X)$ is the distribution function of X, the random variable U is uniformly distributed between 0 and 1. Given n observations $X_{(1)}, \ldots, X_{(n)}$, the values $U_{(i)} = F(X_{(i)})$ are computed. These values are used to compute the EDF test statistics, as described in the next three sections. The CAPABILITY procedures computes the associated *p*-values by interpolating internal tables of probability levels similar to those given by D'Agostino and Stephens (1986).

Kolmogorov-Smirnov Test

The Kolmogorov-Smirnov statistic (D) is defined as

$$D = \sup_x |F_n(x) - F(x)|$$

The Kolmogorov-Smirnov statistic belongs to the supremum class of EDF statistics. This class of statistics is based on the largest vertical difference between $F(x)$ and $F_n(x)$.

The Kolmogorov-Smirnov statistic is computed as the maximum of D^+ and D^-, where D^+ is the largest vertical distance between the EDF and the distribution function when the EDF is greater than the distribution function, and D^- is the largest vertical distance when the EDF is less than the distribution function.

$$\begin{aligned} D^+ &= \max_i \left(\frac{i}{n} - U_{(i)}\right) \\ D^- &= \max_i \left(U_{(i)} - \frac{i-1}{n}\right) \\ D &= \max\left(D^+, D^-\right) \end{aligned}$$

PROC CAPABILITY uses a modified Kolmogorov D statistic to test the data against a normal distribution with mean and variance equal to the sample mean and variance.

Anderson-Darling Test

The Anderson-Darling statistic and the Cramér-von Mises statistic belong to the quadratic class of EDF statistics. This class of statistics is based on the squared difference $(F_n(x) - F(x))^2$. Quadratic statistics have the following general form:

$$Q = n \int_{-\infty}^{+\infty} (F_n(x) - F(x))^2 \, \psi(x) dF(x)$$

The function $\psi(x)$ weights the squared difference $(F_n(x) - F(x))^2$.

The Anderson-Darling statistic (A^2) is defined as

$$A^2 = n \int_{-\infty}^{+\infty} (F_n(x) - F(x))^2 \left[F(x)\left(1 - F(x)\right)\right]^{-1} dF(x)$$

Here the weight function is $\psi(x) = \left[F(x)\left(1 - F(x)\right)\right]^{-1}$.

The Anderson-Darling statistic is computed as

$$A^2 = -n - \frac{1}{n}\sum_{i=1}^{n}\left[(2i-1)\log U_{(i)} + (2n+1-2i)\log\left(\{1 - U_{(i)}\right)\right]$$

Cramér-von Mises Test

The Cramér-von Mises statistic (W^2) is defined as

$$W^2 = n \int_{-\infty}^{+\infty} (F_n(x) - F(x))^2 \, dF(x)$$

Here the weight function is $\psi(x) = 1$.

The Cramér-von Mises statistic is computed as

$$W^2 = \sum_{i=1}^{n}\left(U_{(i)} - \frac{2i-1}{2n}\right)^2 + \frac{1}{12n}$$

Percentile Computations

The CAPABILITY procedure automatically computes the 1st, 5th, 10th, 25th, 50th, 75th, 90th, 95th, and 99th percentiles (quantiles), as well as the minimum and maximum of each analysis variable. To compute percentiles other than these default percentiles, use the PCTLPTS= and PCTLPRE= options in the OUTPUT statement.

You can specify one of five definitions for computing the percentiles with the PCTLDEF= option. Let n be the number of nonmissing values for a variable, and let $x_1, x_2, \ldots, x_n$ represent the ordered values of the variable. Let the t^{th} percentile be y, set $p = \frac{t}{100}$, and let

$$\begin{aligned} np &= j + g \quad \text{when PCTLDEF=1, 2, 3, or 5} \\ (n+1)p &= j + g \quad \text{when PCTLDEF=4} \end{aligned}$$

where j is the integer part of *np*, and g is the fractional part of *np*. Then the PCTLDEF= option defines the t^{th} percentile, y, as described in the following table:

PCTLDEF=	Description	Formula
1	weighted average at x_{np}	$y = (1-g)x_j + gx_{j+1}$ where x_0 is taken to be x_1
2	observation numbered closest to *np*	$y = x_i$ if $g \neq \frac{1}{2}$ $y = x_j$ if $g = \frac{1}{2}$ and *j* is even $y = x_{j+1}$ if $g = \frac{1}{2}$ and *j* is odd where i is the integer part of $np + \frac{1}{2}$
3	empirical distribution function	$y = x_j$ if $g = 0$ $y = x_{j+1}$ if $g > 0$
4	weighted average aimed at $x_{(n+1)p}$	$y = (1-g)x_j + gx_{j+1}$ where x_{n+1} is taken to be x_n
5	empirical distribution function with averaging	$y = \frac{1}{2}(x_j + x_{j+1})$ if $g = 0$ $y = x_{j+1}$ if $g > 0$

Weighted Percentiles

When you use a WEIGHT statement, the percentiles are computed differently. The $100p$th weighted percentile y is computed from the empirical distribution function with averaging

$$y = \begin{cases} \frac{1}{2}(x_i + x_{i+1}) & \text{if } \sum_{j=1}^{i} w_j = pW \\ x_{i+1} & \text{if } \sum_{j=1}^{i} w_j < pW < \sum j = 1^{i+1} w_j \end{cases}$$

where w_i is the weight associated with x_i, and where $W = \sum_{i=1}^{n} w_i$ is the sum of the weights.

Note that the PCTLDEF= option is not applicable when a WEIGHT statement is used. However, in this case, if all the weights are identical, the weighted percentiles are the same as the percentiles that would be computed without a WEIGHT statement and with PCTLDEF=5.

Confidence Limits for Percentiles

You can use the CIPCTLNORMAL option to request confidence limits for percentiles which assume the data are normally distributed. These limits are described in Section 4.4.1 of Hahn and Meeker (1991). When $0.0 < p < 0.5$, the two-sided $100(1-\alpha)\%$ confidence limits for the $100p$-th percentile are

$$\begin{aligned} \text{lower limit} &= \bar{X} - g'(\alpha/2; 1-p, n)s \\ \text{upper limit} &= \bar{X} - g'(1-\alpha/2; p, n)s \end{aligned}$$

where n is the sample size. When $0.5 \leq p < 1.0$, the two-sided $100(1-\alpha)\%$ confidence limits for the $100p$-th percentile are

$$\begin{aligned} \text{lower limit} &= \bar{X} + g'(\alpha/2; 1-p, n)s \\ \text{upper limit} &= \bar{X} + g'(1-\alpha/2; p, n)s \end{aligned}$$

One-sided $100(1-\alpha)\%$ confidence bounds are computed by replacing $\alpha/2$ by α in the appropriate equation above. The factor $g'(\gamma, p, n)$ is related to the noncentral t distribution and is described in Owen and Hua (1977) and Odeh and Owen (1980).

You can use the CIPCTLDF option to request confidence limits for percentiles which are distribution free (in particular, it is not necessary to assume that the data are normally distributed). These limits are described in Section 5.2 of Hahn and Meeker (1991). The two-sided $100(1-\alpha)\%$ confidence limits for the $100p$-th percentile are

$$\begin{aligned} \text{lower limit} &= X_{(l)} \\ \text{upper limit} &= X_{(u)} \end{aligned}$$

where $X_{(j)}$ is the *j*th order statistic when the data values are arranged in increasing order:

$$X_{(1)} \leq X_{(2)} \leq \ldots \leq X_{(n)}$$

The lower rank l and upper rank u are integers that are symmetric (or nearly symmetric) around $[np]+1$ where $[np]$ is the integer part of np, and where n is the sample size. Furthermore, l and u are chosen so that $X_{(l)}$ and $X_{(u)}$ are as close to $X_{[n+1]p}$ as possible while satisfying the coverage probability requirement

$$Q(u-1; n, p) - Q(l-1; n, p) \geq 1-\alpha$$

where $Q(k; n, p)$ is the cumulative binomial probability

$$Q(k; n, p) = \sum_{i=0}^{k} \left(\begin{array}{c} n \\ i \end{array} \right) p^i (1-p)^{n-i}$$

In some cases, the coverage requirement cannot be met, particularly when n is small and p is near 0 or 1. To relax the requirement of symmetry, you can specify CIPCTLDF(TYPE = ASYMMETRIC). This option requests symmetric limits when the coverage requirement can be met, and asymmetric limits otherwise.

If you specify CIPCTLDF(TYPE = LOWER), a one-sided $100(1-\alpha)\%$ lower confidence bound is computed as X_l, where l is the largest integer that satisfies the inequality

$$1 - Q(l-1; n, p) \geq 1-\alpha$$

with $0 < l \leq n$. Likewise, if you specify CIPCTLDF(TYPE = UPPER), a one-sided $100(1-\alpha)\%$ lower confidence bound is computed as X_l, where l is the largest integer that satisfies the inequality

$$Q(u-1; n, p) \geq 1-\alpha$$

where $0 < u \leq n$.

Note that confidence limits for percentiles are not computed when a WEIGHT statement is specified.

Robust Estimators

The CAPABILITY procedure provides several methods for computing robust estimates of location and scale, which are insensitive to outliers in the data.

Winsorized Means

The k-times Winsorized mean is a robust estimator of location which is computed as

$$\bar{x}_{wk} = \frac{1}{n}\left((k+1)x_{(k+1)} + \sum_{i=k+2}^{n-k-1} x_{(i)} + (k+1)x_{(n-k)}\right)$$

where n is the number of observations, and $x_(i)$ is the ith order statistic when the observations are arranged in increasing order:

$$x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(n)}$$

The Winsorized mean is the mean computed after replacing the k smallest observations with the $(k+1)$st smallest observation, and the k largest observations with the $(k+1)$st largest observation.

For data from a symmetric distribution, the Winsorized mean is an unbiased estimate of the population mean. However, the Winsorized mean does not have a normal distribution even if the data are normally distributed.

The Winsorized sum of squared deviations is defined as

$$s^2_{wk} = (k+1)(x_{(k+1)} - \bar{x}_{wk})^2 + \sum_{i=k+2}^{n-k-1}(x_{(i)} - \bar{x}_{wk})^2 + (k+1)(x_{(n-k)} - \bar{x}_{wk})^2$$

A Winsorized t test is given by

$$t_{wk} = \frac{\bar{x}_{wk} - \mu_0}{\mathrm{STDERR}(\bar{\mathrm{x}}_{\mathrm{wk}})}$$

where the standard error of the Winsorized mean is

$$\mathrm{STDERR}(\bar{\mathrm{x}}_{\mathrm{wk}}) = \frac{\mathrm{n}-1}{\mathrm{n}-2\mathrm{k}-1}\frac{\mathrm{s}_{\mathrm{wk}}}{\sqrt{\mathrm{n}(\mathrm{n}-1)}}$$

When the data are from a symmetric distribution, the distribution of t_{wk} is approximated by a Student's t distribution with $n-2k-1$ degrees of freedom. Refer to Tukey and McLaughlin (1963) and Dixon and Tukey (1968).

A $100(1-\alpha)\%$ Winsorized confidence interval for the mean has upper and lower limits

$$\bar{x}_{wk} \pm t_{1-\alpha/2}\mathrm{STDERR}(\bar{\mathrm{x}}_{\mathrm{wk}})$$

where $t_{1-\alpha/2}$ is the $(1-\alpha)/2)100$th percentile of the Student's t distribution with $n-2k-1$ degrees of freedom.

Trimmed Means

The k-times trimmed mean is a robust estimator of location which is computed as

$$\bar{x}_{tk} = \frac{1}{n-2k} \sum_{i=k+1}^{n-k} x_{(i)}$$

where n is the number of observations, and $x_{(i)}$ is the ith order statistic when the observations are arranged in increasing order:

$$x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(n)}$$

The trimmed mean is the mean computed after the k smallest observations and the k largest observations in the sample are deleted.

For data from a symmetric distribution, the trimmed mean is an unbiased estimate of the population mean. However, the trimmed mean does not have a normal distribution even if the data are normally distributed.

A robust estimate of the variance of the trimmed mean t_{tk} can be obtained from the Winsorized sum of squared deviations; refer to Tukey and McLaughlin (1963). the corresponding trimmed t test is given by

$$t_{tk} = \frac{\bar{x}_{tk} - \mu_0}{\mathrm{STDERR}(\bar{\mathrm{x}}_{\mathrm{tk}})}$$

where the standard error of the trimmed mean is

$$\mathrm{STDERR}(\bar{\mathrm{x}}_{\mathrm{tk}}) = \frac{\mathrm{s_{tk}}}{\sqrt{(\mathrm{n}-2\mathrm{k})(\mathrm{n}-2\mathrm{k}-1)}}$$

and s_{wk} is the square root of the Winsorized sum of squared deviations.

When the data are from a symmetric distribution, the distribution of t_{tk} is approximated by a Student's t distribution with $n-2k-1$ degrees of freedom. Refer to Tukey and McLaughlin (1963) and Dixon and Tukey (1968).

A $100(1-\alpha)\%$ trimmed confidence interval for the mean has upper and lower limits

$$\bar{x}_{tk} \pm t_{1-\alpha/2} \mathrm{STDERR}(\bar{\mathrm{x}}_{\mathrm{tk}})$$

where $t_{1-\alpha/2}$ is the $(1-\alpha)/2)100$th percentile of the Student's t distribution with $n-2k-1$ degrees of freedom.

Robust Estimates of Scale

The sample standard deviation, which is the most commonly used estimator of scale, is sensitive to outliers. Robust scale estimators, on the other hand, remain bounded when a single data value is replaced by an arbitrarily large or small value. The CAPABILITY procedure computes several robust measures of scale, including the interquartile range Gini's mean difference G, the median absolute deviation about the median (MAD), Q_n, and S_n. In addition, the procedure computes estimates of the normal standard deviation σ derived from each of these measures.

The interquartile range (IQR) is simply the difference between the upper and lower quartiles. For a normal population, σ can be estimated as IQR/1.34898.

Gini's mean difference is computed as

$$G = \frac{1}{\left(\begin{array}{c} n \\ 2 \end{array} \right)} \sum_{i<j} |x_i - x_j|$$

For a normal population, the expected value of G is $2\sigma/\sqrt{\pi}$. Thus $G\sqrt{pi}/2$ is a robust estimator of σ when the data are from a normal sample. For the normal distribution, this estimator has high efficiency relative to the usual sample standard deviation, and it is also less sensitive to the presence of outliers.

A very robust scale estimator is the MAD, the median absolute deviation from the median (Hampel, 1974), which is computed as

$$\mathrm{MAD} = \mathrm{med_i}(|\mathrm{x_i} - \mathrm{med_j}(\mathrm{x_j})|)$$

where the inner median, $\mathrm{med_j}(\mathrm{x_j})$, is the median of the n observations, and the outer median (taken over i) is the median of the n absolute values of the deviations about the inner median. For a normal population, 1.4826MAD is an estimator of σ.

The MAD has low efficiency for normal distributions, and it may not always be appropriate for symmetric distributions. Rousseeuw and Croux (1993) proposed two statistics as alternatives to the MAD. The first is

$$S_n = 1.1926\mathrm{med_i}(\mathrm{med_j}(|\mathrm{x_i} - \mathrm{x_j}|))$$

where the outer median (taken over i) is the median of the n medians of $|x_i - x_j|$, $j = 1, 2, \ldots, n$. To reduce small-sample bias, $c_{sn}S_n$ is used to estimate σ, where c_{sn} is a correction factor; refer to Croux and Rousseeuw (1992).

The second statistic is

$$Q_n = 2.219\{|x_i - x_j|; i < j\}_{(k)}$$

where

$$k = \left(\begin{array}{c} h \\ 2 \end{array} \right)$$

and $h = [n/2] + 1$. In other words, Q_n is 2.219 times the kth order statistic of the $\binom{n}{2}$ distances between the data points. The bias-corrected statistic $c_{qn}Q_n$ is used to estimate σ, where c_{qn} is a correction factor; refer to Croux and Rousseeuw (1992).

Computing the Mode

The mode is the value that occurs most often in a set of observations. The CAPABILITY procedure counts repetitions of the actual values (or the rounded values, if you specify the ROUND= option). If a tie occurs for the most frequent value, the procedure reports the lowest mode in the table labeled *Basic Statistical Measures*. To list all possible modes, specify the MODES option in the PROC CAPABILITY statement. When no repetitions occur in the data, the procedure does not report the mode. The WEIGHT statement has no effect on the mode.

Assumptions and Terminology for Capability Indices

One of the fundamental assumptions in process capability analysis is that the process must be in statistical control. Without statistical control, the process is not predictable, the concept of a process distribution does not apply, and quantities related to the distribution, such as probabilities, percentiles, and capability indices, cannot be meaningfully estimated. Additionally, all of the standard process capability indices described in the next section require that the process distribution be normal, or at least approximately normal.

In many industries, statistical control is routinely checked with a Shewhart chart (such as an $\bar{X}$ and R chart) before capability indices such as

$$C_{pk} = \min\left(\frac{\text{USL} - \mu}{3\sigma}, \frac{\text{LSL} - \mu}{3\sigma}\right)$$

are computed. The control chart analysis yields estimates for the process mean μ and standard deviation σ, which are based on subgrouped data and can be used to estimate C_{pk}. In particular, σ can be estimated by

$$s_R = \bar{R}/d_2$$

rather than the ungrouped sample standard deviation

$$s = \frac{1}{n-1}\sqrt{\sum_{i=1} n(x_i - \bar{x})/^2}$$

You can use the SHEWHART procedure to carry out the control chart analysis and to compute capability indices based on s_R. On the other hand, the CAPABILITY procedure computes indices based on s.

Some industry manuals distinguish these two approaches. For instance, the ASQC/AIAG manual *Fundamental Process Control* uses the notation C_{pk} for the estimate based on s_R, and it uses the notation P_{pk} for the estimate based on s. However,

assuming that the process is in control and only common cause variation is present, both s_R and s are estimates of the same parameter σ, and so there is fundamentally no difference in the two approaches*.

Once control has been established, attention should focus on the distribution of the process measurements, and at this point there is no practical or statistical advantage to working with subgrouped measurements. In fact, the use of s is closely associated with a wide variety of methods that are highly useful for process capability analysis, including tests for normality, graphical displays such as histograms and probability plots, and confidence intervals for parameters and capability indices.

Standard Capability Indices

This section provides computational details for the standard process capability indices computed by the CAPABILITY procedure: C_p, *CPL*, *CPU*, C_{pk}, and C_{pm}.

The Index Cp

The process capability index C_p, sometimes called the "process potential index," the "process capability ratio," or the "inherent capability index," is estimated as

$$\widehat{C}_p = \frac{USL - LSL}{6s}$$

where *USL* is the upper specification limit, *LSL* is the lower specification limit, and s is the sample standard deviation. If you do not specify both the upper and the lower specification limits in the SPEC statement or the SPEC= data set, then C_p is assigned a missing value.

The interpretation of C_p can depend on the application, on past experience, and on local practice. However, broad guidelines for interpretation have been proposed by several authors. Ekvall and Juran (1974) classify C_p values as

- "not adequate" if $C_p < 1$
- "adequate" if $1 \leq C_p \leq 1.33$, but requiring close control as C_p approaches 1
- "more than adequate" if $C_p > 1.33$

Montgomery (1996) recommends minimum values of C_p as

- 1.33 for existing processes
- 1.50 for new processes or for existing processes when the variable is critical (for example, related to safety or strength)
- 1.67 for new processes when the variable is critical

*Statistically, s is a more efficient estimator of σ than s_R.

Exact $100(1-\alpha)\%$ lower and upper confidence limits for C_p (denoted by LCL and UCL) are computed using percentiles of the chi-square distribution, as indicated by the following equations:

$$\begin{aligned} \text{lower limit} &= \hat{C}_p\sqrt{\chi^2_{\alpha/2,n-1}/(n-1)} \\ \text{upper limit} &= \hat{C}_p\sqrt{\chi^2_{1-\alpha/2,n-1}/(n-1)} \end{aligned}$$

Here, $\chi^2_{\alpha,\nu}$ denotes the lower $100\alpha^{\text{th}}$ percentile of the chi-square distribution with ν degrees of freedom. Refer to Chou et al. (1990) and Kushler and Hurley (1992).

You can specify α with the ALPHA= option in the PROC CAPABILITY statement or with the CIINDICES(ALPHA=*value*) in the PROC CAPABILITY statement. The default *value* is 0.05. You can save these limits in the OUT= data set by specifying the keywords CPLCL and CPUCL in the OUTPUT statement. In addition, you can display these limits on plots produced by the CAPABILITY procedure by specifying the keywords in the INSET statement.

The Index CPL

The process capability index *CPL* is estimated as

$$\widehat{\text{CPL}} = \frac{\bar{X} - LSL}{3s}$$

where $\bar{X}$ is the sample mean, *LSL* is the lower specification limit, and s is the sample standard deviation. If you do not specify the lower specification limit in the SPEC statement or the SPEC= data set, then *CPL* is assigned a missing value.

Montgomery (1996) refers to *CPL* as the "process capability ratio" in the case of one-sided lower specifications and recommends minimum values as follows:

- 1.25 for existing processes
- 1.45 for new processes or for existing processes when the variable is critical
- 1.60 for new processes when the variable is critical

Exact $100(1-\alpha)\%$ lower and upper confidence limits for *CPL* are computed using a generalization of the method of Chou et al. (1990), who point out that the $100(1-\alpha)$ lower confidence limit for *CPL* (denoted by CPLLCL)satisfies the equation

$$\Pr\{T_{n-1}(\delta = 3\sqrt{n})\ \text{CPLLCL}\ \leq 3\text{CPL}\sqrt{n}\} = 1-\alpha$$

where $T_{n-1}(\delta)$ has a non-central t distribution with $n-1$ degrees of freedom and noncentrality parameter δ. You can specify α with the ALPHA= option in the PROC CAPABILITY statement. The default value is 0.05. The confidence limits can be saved in an output data set by specifying the keywords CPLLCL and CPLUCL in the OUTPUT statement. In addition, you can display these limits on plots produced by the CAPABILITY procedure by specifying these keywords in the INSET statement.

The Index CPU

The process capability index *CPU* is estimated as

$$\widehat{\mathrm{CPU}} = \frac{USL - \bar{X}}{3s}$$

where *USL* is the upper specification limit, $\bar{X}$ is the sample mean, and s is the sample standard deviation. If you do not specify the upper specification limit in the SPEC statement or the SPEC= data set, then *CPU* is assigned a missing value.

Montgomery (1996) refers to *CPU* as the "process capability ratio" in the case of one-sided upper specifications and recommends minimum values that are the same as those specified previously for *CPL*.

Exact $100(1-\alpha)\%$ lower and upper confidence limits for *CPU* are computed using a generalization of the method of Chou et al. (1990), who point out that the $100(1-\alpha)$ lower confidence limit for *CPU* (denoted by CPULCL)satisfies the equation

$$\Pr\{T_{n-1}(\delta = 3\sqrt{n}\,\mathrm{CPULCL}\ \geq 3\mathrm{CPU}\sqrt{n}\} = 1-\alpha$$

where $T_{n-1}(\delta)$ has a non-central t distribution with $n-1$ degrees of freedom and noncentrality parameter δ. You can specify α with the ALPHA= option in the PROC CAPABILITY statement. The default value is 0.05. The confidence limits can be saved in an output data set by specifying the keywords CPULCL and CPUUCL in the OUTPUT statement. In addition, you can display these limits on plots produced by the CAPABILITY procedure by specifying these keywords in the INSET statement.

The Index Cpk

The process capability index C_{pk} is defined as

$$C_{pk} = \frac{1}{3\sigma}\min(USL - \mu,\ \mu - LSL) = \min(CPU,\ CPL)$$

Note that the indices C_{pk}, C_p, and k are related as $C_{pk} = C_p(1-k)$. The CAPABILITY procedure estimates C_{pk} as

$$\widehat{C}_{pk} = \frac{1}{3s} \times \min(USL - \bar{X},\ \bar{X} - LSL) = \min(CPU,\ CPL)$$

where *USL* is the upper specification limit, *LSL* is the lower specification limit, $\bar{X}$ is the sample mean, and s is the sample standard deviation.

If you specify only the upper limit in the SPEC statement or the SPEC= data set, then C_{pk} is computed as *CPU*, and if you specify only the lower limit in the SPEC statement or the SPEC= data set, then C_{pk} is computed as *CPL*.

Bissell (1990) derived approximate two-sided 95% confidence limits for C_{pk} by assuming that the distribution of $\widehat{C}_{pk}$ is normal. Using Bissell's approach, $100(1-\alpha)\%$ lower and upper confidence limits can be computed as

$$\text{lower limit} = \widehat{C}_{pk}\left[1 - \Phi^{-1}(1-\alpha/2)\sqrt{\frac{1}{9n\widehat{C}_{pk}^2} + \frac{1}{2(n-1)}}\right]$$

$$\text{upper limit} = \widehat{C}_{pk}\left[1 + \Phi^{-1}(1-\alpha/2)\sqrt{\frac{1}{9n\widehat{C}_{pk}^2} + \frac{1}{2(n-1)}}\right]$$

where Φ denotes the cumulative standard normal distribution function. Kushler and Hurley (1992) concluded that Bissell's method gives reasonably accurate results.

You can specify α with the ALPHA= option in the PROC CAPABILITY statement. The default value is 0.05. These limits can be saved in an output data set by specifying the keywords CPKLCL and CPKUCL in the OUTPUT statement. In addition, you can display these limits on plots produced by the CAPABILITY procedure by specifying these same keywords in the INSET statement.

The Index Cpm

The process capability index C_{pm} is intended to account for deviation from the target T in addition to variability from the mean. This index is often defined as

$$C_{pm} = \frac{\text{USL} - \text{LSL}}{6\sqrt{\sigma^2 + (\mu - T)^2}}$$

A closely related version of C_{pm} is the index

$$C_{pm}^* = \frac{\min(\text{USL} - \text{T}, \text{T} - \text{LSL})}{3\sqrt{\sigma^2 + (\mu - T)^2}} = \frac{d - |T - m|}{3\sqrt{\sigma^2 + (\mu - T)^2}}$$

where $d = (\text{USL} - \text{LSL})/2$ and $m = (\text{USL} + \text{LSL})/2$. If $T = m$, then $C_{pm} = C_{pm}^*$. However, if $T \neq m$, then both indices suffer from problems of interpretation, as pointed out by Kotz and Johnson (1993), and their use should be avoided in this case.

The CAPABILITY procedure computes an estimator of C_{pm} as

$$\widehat{C}_{pm} = \frac{\min(\mathit{USL} - T,\ T - \mathit{LSL})}{3\sqrt{s^2 + (\bar{X} - T)^2}}$$

where s is the sample standard deviation.

If you specify only a single specification limit *SL* in the SPEC statement or the SPEC= data set, then C_{pm} is estimated as

$$\widehat{C}_{pm} = \frac{|T - \mathit{SL}|}{3\sqrt{s^2 + (\bar{X} - T)^2}}$$

Boyles (1991) proposed a slightly modified point estimate for C_{pm} computed as

$$\widetilde{C}_{pm} = \frac{(\text{USL} - \text{LSL})/2}{3\sqrt{(\frac{n-1}{n})s^2 + (\bar{X} - T)^2}}$$

Boyles also suggested approximate two-sided $100(1-\alpha)\%$ confidence limits for C_{pm}, which are computed as

$$\begin{aligned} \text{lower limit} &= \widetilde{C}_{pm}\sqrt{\chi^2_{\alpha/2,\nu}/\nu} \\ \text{upper limit} &= \widetilde{C}_{pm}\sqrt{\chi^2_{1-\alpha/2,\nu}/\nu} \end{aligned}$$

Here $\chi^2_{\alpha,\nu}$ denotes the lower $100\alpha^{\text{th}}$ percentile of the chi-square distribution with ν degrees of freedom, where ν equals

$$\frac{n(1 + (\frac{\bar{X}-T}{s})^2)}{1 + 2(\frac{\bar{X}-T}{s})^2}$$

You can specify α with the ALPHA= option in the PROC CAPABILITY statement. The default value is 0.05. These confidence limits can be saved in an output data set by specifying the keywords CPMLCL and CPMUCL in the OUTPUT statement. In addition, you can display these limits on plots produced by the CAPABILITY procedure by specifying these keywords in the INSET statement.

Specialized Capability Indices

This section describes a number of specialized capability indices which you can request with the SPECIALINDICES option in the PROC CAPABILITY statement.

The Index k

The process capability index k (also denoted by *K*) is computed as

$$k = \frac{2|m - \bar{X}|}{USL - LSL}$$

where $m = \frac{1}{2}(USL + LSL)$ is the midpoint of the specification limits, $\bar{X}$ is the sample mean, *USL* is the upper specification limit, and *LSL* is the lower specification limit.

The formula for k used here is given by Kane (1986). Note that k is sometimes computed without taking the absolute value of $m - \bar{X}$ in the numerator. See Wadsworth *et al.* (1986).

If you do not specify the upper and lower limits in the SPEC statement or the SPEC= data set, then k is assigned a missing value.

Boyles' Index C_{pm}^{+}

Boyles (1992) proposed the process capability index C_{pm}^{+} which is defined as

$$C_{pm}^{+} = \frac{1}{3}\left[\frac{E_{X<T}\left[(X-T)^2\right]}{(T-\text{LSL})^2} + \frac{E_{X>T}\left[(X-T)^2\right]}{(\text{USL}-\text{T})^2}\right]^{-1/2}$$

He proposed this index as a modification of C_{pm} for use when $\mu \neq T$. The quantities

$$E_{X<T}\left[(X-T)^2\right] = E\left[(X-T)^2 | X<T\right] Pr\left[X<T\right]$$

and

$$E_{X>T}\left[(X-T)^2\right] = E\left[(X-T)^2 | X>T\right] Pr\left[X>T\right]$$

are referred to as semivariances. Kotz and Johnson (1993) point out that if $T = (\text{LSL}+\text{USL})/2$, then $C_{pm}^{+} = C_{pm}$.

Kotz and Johnson (1993) suggest that a natural estimator for C_{pm}^{+} is

$$\widehat{C}_{pm}^{+} = \frac{1}{3}\left[\frac{1}{n}\left\{\frac{\sum_{X_i<T}(X_i-T)^2}{(T-\text{LSL})^2} + \frac{\sum_{X_i>T}(X_i-T)^2}{(\text{USL}-T)^2}\right\}^{-1/2}\right]$$

Note that this index is not defined when either of the specification limits is equal to the target T. Refer to Section 3.5 of Kotz and Johnson (1993) for further details.

The Index C_{jkp}

Johnson, Kotz, and Pearn (1994) introduced a so-called "flexible" process capability index which takes into account possible differences in variability above and below the target T. They defined this index as

$$C_{jkp} = \frac{1}{3\sqrt{2}}\min\left(\frac{\text{USL}-T}{\sqrt{E_{X>T}[(X-T)^2]}}, \frac{T-\text{LSL}}{\sqrt{E_{X<T}[(X-T)^2]}}\right)$$

where $d = (\text{USL}-\text{LSL})/2$.

A natural estimator of this index is

$$\widehat{C}_{jkp} = \frac{1}{3\sqrt{2}}\min\left(\frac{\text{USL}-T}{\sqrt{\sum_{X_i>T}(X_i-T)^2/n}}, \frac{T-\text{LSL}}{\sqrt{\sum_{X_i<T}(X_i-T)^2/n}}\right)$$

For further details, refer to Section 4.4 of Kotz and Johnson (1993).

The Indices $C_{pm}(a)$

The class of capability indices $C_{pm}(a)$, indexed by the parameter a $(a > 0)$ allows flexibility in choosing between the relative importance of variability and deviation of the mean from the target value T.

The class defined as

$$C_{pm}(a) = (1 - a\zeta^2)C_p$$

where $\zeta = (\mu - T)/\sigma$. The motivation for this definition is that if $|\zeta|$ is small, then

$$C_{pm} \approx (1 - \frac{1}{2}\zeta^2)C_p$$

A natural estimator of $C_{pm}(a)$ is

$$\frac{d}{3s}\widehat{C}_{pm}(a) = \left\{1 - a\left(\frac{\bar{X} - T}{s}\right)^2\right\}$$

where $d = (\text{USL} - \text{LSL})/2$. You can specify the value of a with the SPECIALINDICES(CPMA=) option in the PROC CAPABILITY statement. By default, $a = 0.5$.

This index is not recommended for situation in which the target T is not equal to the midpoint of the specification limits.

For additional details, refer to Section 3.7 of Kotz and Johnson (1993).

The Index $C_{p(5.15)}$

Johnson *et al.* (1992) suggest the class of process capability indices defined as

$$C_{p(\theta)} = \frac{\text{USL} - \text{LSL}}{\theta\sigma}$$

where θ is chosen so that the proportion of conforming items is robust with respect to the shape of the process distribution. In particular, Kotz and Johnson (1993) recommend use of

$$C_{p(5.15)} = \frac{\text{USL} - \text{LSL}}{5.15\sigma}$$

which is estimated as

$$\widehat{C}_{p(5.15)} = \frac{\text{USL} - \text{LSL}}{5.15s}$$

For details, refer to Section 4.3.2 of Kotz and Johnson (1993).

The Index $C_{pk(5.15)}$

Similarly, Kotz and Johnson (1993) recommend use of the robust capability index

$$C_{pk(5.15)} = \frac{d - |\mu - (\text{USL} + \text{LSL})/2|}{2.575\sigma}$$

where $d = (\text{USL} - \text{LSL})/2$. This index is estimated as

$$\widehat{C}_{pk(5.15)} = \frac{d - |\bar{X} - (\text{USL} + \text{LSL})/2|}{2.575s}$$

For details, refer to Section 4.3.2 of Kotz and Johnson (1993).

The Index C_{pmk}

Pearn *et al.* (1992) proposed the index C_{pmk}

$$C_{pmk} = \frac{(\text{USL} - \text{LSL})/2 - |\mu - m|}{3\sqrt{\sigma^2 + (\mu - T)^2}}$$

where $m = (\text{LCL} + \text{UCL})/2$. A natural estimator for C_{pmk} is

$$\widehat{C}_{pmk} = \frac{(\text{USL} - \text{LSL})/2 - |\bar{X} - m|}{3\sqrt{(\frac{n-1}{n})s^2 + (\bar{X} - T)^2}}$$

where $m = (\text{USL} + \text{LSL})/2$.

For further details, refer to Section 3.6 of Kotz and Johnson (1993).

Wright's Index C_s

Wright (1995) defines the capability index

$$C_s = \frac{\min(\text{USL} - \mu, \mu - \text{LSL})}{3\sqrt{\sigma^2 + (\mu - T)^2 + \mu_3/\sigma}}$$

where $\mu_3 = E(X - \mu)^3$.

A natural estimator of C_s is

$$\widehat{C}_s = \frac{(\text{USL} - \text{LSL})/2 - |\bar{\text{X}} - \text{m}|}{3\sqrt{\left(\frac{n-1}{n}\right)s^2 + (\bar{X} - T)^2 + |c_4 s^2 b_3|}}$$

where c_4 is an unbiasing constant for the sample standard deviation, and b_3 is a measure of skewness. Wright (1995) shows that C_s compares favorably with C_{pmk} even when skewness is not present, and he advocates the use of C_s for monitoring near-normal processes when loss of capability typically leads to asymmetry.

Chen and Kotz (1996) proposed a modification to Wright's C_s index which introduces a multiplier, $\gamma > 0$, and is estimated as

$$\widehat{C}_s = \frac{(\text{USL} - \text{LSL})/2 - |\bar{\text{X}} - \text{m}|}{3\sqrt{\left(\frac{n-1}{n}\right) s^2 + (\bar{X} - T)^2 + \gamma |c_4 s^2 b_3|}}$$

If you specify a value for γ with the SPECIALINDICES(CSGAMMA=) option, the index C_s is computed with this modification. Otherwise it is computed using Wright's original definition.

The Index S_{jkp}

Boyles (1994) proposed a smooth version of C_{jkp} defined as

$$S_{jkp} = S\left(\frac{\text{USL} - T}{\sqrt{2E_{X>T}[(X-T)^2]}}, \frac{T - \text{LSL}}{\sqrt{2E_{X<T}[(X-T)^2]}}\right)$$

The CAPABILITY procedure estimates S_{jkp} as

$$\widehat{S}_{jkp} = S\left(\frac{\text{USL} - T}{\sqrt{2\sum_{X_i>T}(X_i - T)^2/n}}, \frac{T - \text{LSL}}{\sqrt{2\sum_{X_i<T}(X_i - T)^2/n}}\right)$$

where $S(x, y) = \Phi^{-1}[\{\Phi(x) + \Phi(y)\}/2]/3$.

The Index C_{pp}

Chen (1998) devised a process incapability index based on the C^*_{pm} index. The first term measures *inaccuracy* and the second measures *imprecision*. The C_{pp} index is estimated as

$$\widehat{C}_{pp} = \left(\frac{\bar{X} - T}{d^*/3}\right)^2 + \left(\frac{s}{d^*/3}\right)^2$$

where $d^* = \min(\text{USL} - T, T - \text{LSL})$.

The Index C''_{pp}

The index C_{pp} does not handle asymmetric tolerances well, as discussed by Kotz and Lovelace (1998). To address that shortcoming, Chen (1998) defined the index C''_{pp}, which is estimated by

$$\widehat{C}''_{pp} = \left(\frac{\widehat{A}}{d^*/3}\right)^2 + \left(\frac{s}{d^*/3}\right)$$

where

$$\widehat{A} = \max\left\{\frac{(\bar{X}-T)d}{T-\text{LSL}}, \frac{(T-\bar{X})d}{\text{USL}-T}\right\}$$

and $d = (USL - LSL)/2$.

The Index C_{pg}

Marcucci and Beazley (1988) defined the index

$$C_{pg} = \frac{1}{C_{pm}^2}$$

which is estimated as

$$\widehat{C}_{pg} = \frac{1}{\widehat{C}_{pm}^2}$$

The Index C_{pq}

Gupta and Kotz (1997) introduced the index C_{pq}, which is estimated by

$$\widehat{C}_{pq} = \widehat{C}_p\left[1 - \frac{1}{2}\left(\frac{\bar{X}-T}{s}\right)^2\right]$$

The Index C_p^W

Bai and Choi (1997) defined the index

$$C_p^W = \frac{C_p}{\sqrt{1+|1-2P_x|}}$$

where $P_x = \Pr(X \leq \mu)$. It is estimated by

$$\widehat{C}_p^W = \frac{\widehat{C}_p}{\sqrt{1+|1-2\widehat{P}_x|}}$$

where $\widehat{P}_x$ is the fraction of observations less than or equal to $\bar{X}$. For more information on C_p^W, see Kotz and Lovelace (1998).

The Index C_{pk}^W

Bai and Choi (1997) also proposed the index

$$C_{pk}^W = \min\left\{\frac{\text{USL}-\mu}{3\sigma\sqrt{2P_x}}, \frac{\mu-\text{LSL}}{3\sigma\sqrt{2(1-P_x)}}\right\}$$

It is estimated by

$$\widehat{C}_{pk}^W = \min\left\{\frac{\text{USL}-\bar{X}}{3s\sqrt{2\widehat{P}_x}}, \frac{\bar{X}-\text{LSL}}{3s\sqrt{2(1-\widehat{P}_x)}}\right\}$$

where $\widehat{P}_x$ is the fraction of observations less than or equal to $\bar{X}$. For more information on C_{pk}^W, see Kotz and Lovelace (1998).

The Index C_{pm}^W

The index C_{pm}^W, also introduced by Bai and Choi (1997), is defined as

$$C_{pm}^W = \frac{C_{pm}}{\sqrt{1+|1-2P_T|}}$$

where $P_T = \Pr(X \le T)$. It is estimated by

$$\widehat{C}_{pm}^W = \frac{\widehat{C}_{pm}}{\sqrt{1+|1-2\widehat{P}_T|}}$$

where $\widehat{P}_T$ is the fraction of observations less than or equal to T. For more information on C_{pm}^W, see Kotz and Lovelace (1998).

The Index C_{pc}

Luceño (1996) proposed the index

$$C_{pc} = \frac{\text{USL}-\text{LSL}}{6\sqrt{\frac{\pi}{2}E|X-M|}}$$

where $M = (\text{USL}+\text{LSL})/2$. It is estimated by

$$\widehat{C}_{pc} = \frac{\text{USL}-\text{LSL}}{6\sqrt{\frac{\pi}{2}c}}$$

where

$$c = \frac{1}{n}\sum_{i=1}^{n}|X_i - M|$$

Vännmann's Index $C_p(u, v)$

Vännmann (1995) introduced the generalized index $C_p(u, v)$, which reduces to the following capability indices given appropriate choices of u and v:

- $C_p(0, 0) = C_p$
- $C_p(0, 1) = C_{pk}$
- $C_p(1, 0) = C_{pm}$
- $C_p(1, 1) = C_{pmk}$

$C_p(u, v)$ is defined as

$$C_p(u, v) = \frac{d - u|\ \mu - M|}{3\sqrt{\sigma^2 + v(\ \mu - T)^2}}$$

and estimated by

$$\widehat{C}_p(u, v) = \frac{d - u|\bar{X} - M|}{3\sqrt{(\frac{n-1}{n})s^2 + v(\bar{X} - T)^2}}$$

You can specify u with the SPECIALINDICES(CPU=) option and v with the SPECIALINDICES(CPV=) option. By default, $u = 0$ and $v = 4$.

Vännmann's Index $C_p(v)$

Vännmann (1997) also proposed the index $C_p(v)$, which is equivalent to $C_p(u, v)$ with $u = 1$. It is estimated as

$$\widehat{C}_p(v) = \frac{d - |\bar{X} - M|}{3\sqrt{(\frac{n-1}{n})s^2 + v(\bar{X} - T)^2}}$$

You can specify v with the SPECIALINDICES(CPV=) option. By default, $v = 4$.

Missing Values

If a variable for which statistics are calculated has a missing value, that value is ignored in the calculation of statistics, and the missing values are tabulated separately. A missing value for one such variable does not affect the treatment of other variables in the same observation.

If the WEIGHT variable has a missing value, the observation is excluded from the analysis*. If the FREQ variable has a missing value, the observation is excluded from the analysis. If a variable in a BY or ID statement has a missing value, the procedure treats it as it would treat any other value of a BY or ID variable.

*In Release 6.12 and earlier releases, the observation was included in the analysis, and the missing WEIGHT variable value was taken to be zero.

ODS Tables

This section describes the ODS tables produced by the CAPABILITY procedure.

The following table summarizes the ODS tables that you can request with options in the PROC CAPABILITY statement.

Table 8.21. ODS Tables Produced with the PROC CAPABILITY Statement

Table Name	Description	Option
BasicIntervals	confidence intervals for mean, standard deviation, variance	CIBASIC
BasicMeasures	measures of location and variability	default
ExtremeObs	extreme observations	default
ExtremeValues	extreme values	NEXTRVAL=
Frequencies	frequencies	FREQ
LocationCounts	counts used for sign test and signed rank test	LOCCOUNTS
MissingValues	missing values	default
Modes	modes	MODES
Moments	sample moments	default
Quantiles	quantiles	default
RobustScale	robust measures of scale	ROBUSTSCALE
TestsForLocation	tests for location	default
TestsForNormality	tests for normality	NORMALTEST
TrimmedMeans	trimmed means	TRIMMED=
WinsorizedMeans	Winsorized means	WINSORIZED=

The following table summarizes the ODS tables related to capability indices that you can request with options in the PROC CAPABILITY statement when you provide specification limits with a SPEC statement or with a SPEC= data set.

Table 8.22. ODS Tables Related to Specification Limits

Table Name	Description	Option
CIProbExSpecs	confidence limits for probabilities of exceeding specifications	CIPROBEX in PROC CAPABILITY statement
Indices	standard capability indices	default
SpecialIndices	specialized capability indices	SPECIALINDICES in PROC CAPABILITY statement
Specifications	percents outside specification limits based on empirical	default

The following table summarizes the ODS tables related to fitted distributions that you can request with options in the HISTOGRAM statement.

Table 8.23. ODS Tables Produced with the HISTOGRAM Statement

Table Name	Description	Option
Bins	histogram bins	MIDPERCENTS sub-option with any distribution option, such as NORMAL(MIDPERCENTS)
FitIndices	capability indices computed from fitted distribution	INDICES sub-option with any distribution option, such as LOGNORMAL(INDICES)
FitQuantiles	quantiles of fitted distribution	any distribution option such as NORMAL
GoodnessOfFit	goodness-of-fit tests for fitted distribution	any distribution option such as NORMAL
ParameterEstimates	parameter estimates for fitted distribution	any distribution option such as NORMAL
Specifications	percents outside specification limits based on empirical and fitted distributions	any distribution option such as NORMAL

The following table summarizes the ODS tables that you can request with options in the INTERVALS statement.

Table 8.24. ODS Tables Produced with the INTERVALS Statement

Table Name	Description	Option
Intervals1	prediction interval for future observations	METHODS=1
Intervals2	prediction interval for mean	METHODS=2
Intervals3	tolerance interval for proportion of population	METHODS=3
Intervals4	confidence limits for mean	METHODS=4
Intervals5	prediction interval for standard deviation	METHODS=5
Intervals6	confidence limits for standard deviation	METHODS=6

Examples

This section provides a more advanced example of the PROC CAPABILITY statement.

Example 8.1. Reading Specification Limits

See CAPSPEC2 in the SAS/QC Sample Library

You can specify specification limits either in the SPEC statement or in a SPEC= data set. In "Computing Capability Indices" on page 169, limits were specified in a SPEC statement. This example illustrates how to create a SPEC= data set to read specification limits with the SPEC= option in the PROC CAPABILITY statement.

Consider the drink can data presented in "Computing Descriptive Statistics" on page 166. Suppose, in addition to the fluid weight of each drink can, the weight of the can itself is stored in a variable named CWEIGHT, and both variables are saved in a data set called CAN2. A partial listing of CAN2 follows:

Output 8.1.1. The Data Set CAN2

Obs	weight	cweight
1	12.07	1.07
2	12.02	0.86
3	12.00	1.06
.	.	.
.	.	.
.	.	.
100	11.98	1.07

The following data step creates a data set named LIMITS containing specification limits for the fluid weight and the can weight. LIMITS has 4 variables (_VAR_, _LSL_, _USL_, and _TARGET_) and 2 observations. The first observation contains the specification limit information for the variable WEIGHT, and the second contains the specification limit information for the variable CWEIGHT.

```
data limits;
  length _var_ $8;
  _var_    = 'weight';
  _lsl_    = 11.95;
  _target_ = 12;
  _usl_    = 12.05;
  output;
  _var_    = 'cweight';
  _lsl_    = 0.90;
  _target_ = 1;
  _usl_    = 1.10;
  output;
run;
```

The following statements read the specification information from the LIMITS data set into the CAPABILITY procedure using the SPEC= option. These statements

print summary statistics, capability indices, and specification limit information for WEIGHT and CWEIGHT. Figure 8.1 (page 168) and Figure 8.2 (page 169) display the output for WEIGHT. Output 8.1.2 displays the output for CWEIGHT.

```
title 'Process Capability Analysis of Drink Can Data';
proc capability data=can2 specs=limits;
   var cweight;
run;
```

Output 8.1.2. Printed Output for Variable CWEIGHT

```
            Process Capability Analysis of Drink Can Data

              Variable:  cweight  (Can Weight (ounces))

                              Moments

N                          100    Sum Weights               100
Mean                     1.004    Sum Observations        100.4
Std Deviation      0.06330941    Variance            0.00400808
Skewness            -0.074821    Kurtosis            -0.5433858
Uncorrected SS       101.1984    Corrected SS            0.3968
Coeff Variation    6.30571767    Std Error Mean      0.00633094


                    Basic Statistical Measures

          Location                      Variability

      Mean     1.004000     Std Deviation            0.06331
      Median   1.000000     Variance                 0.00401
      Mode     1.040000     Range                    0.29000
                            Interquartile Range      0.08500

NOTE: The mode displayed is the smallest of 2 modes with a count of 8.


                    Tests for Location: Mu0=0

          Test            -Statistic-    -----p Value------

          Student's t     t  158.5862    Pr > |t|    <.0001
          Sign            M        50    Pr >= |M|   <.0001
          Signed Rank     S      2525    Pr >= |S|   <.0001


                       Tests for Normality

      Test                  --Statistic---    -----p Value-----

      Shapiro-Wilk          W     0.987310    Pr < W      0.459
      Kolmogorov-Smirnov    D     0.061410    Pr > D     >0.150
      Cramer-von Mises      W-Sq  0.048175    Pr > W-Sq  >0.250
      Anderson-Darling      A-Sq  0.361939    Pr > A-Sq  >0.250
```

Output 8.1.3. Printed Output for Variable CWEIGHT (cont.)

Process Capability Analysis of Drink Can Data

Variable: cweight (Can Weight (ounces))

Quantiles (Definition 5)

Quantile	Estimate
100% Max	1.150
99%	1.140
95%	1.105
90%	1.080
75% Q3	1.045
50% Median	1.000
25% Q1	0.960
10%	0.910
5%	0.900
1%	0.870
0% Min	0.860

Extreme Observations

----Lowest----		----Highest---	
Value	Obs	Value	Obs
0.86	2	1.11	42
0.88	89	1.12	28
0.88	64	1.12	34
0.90	68	1.13	48
0.90	59	1.15	52

Specification Limits

--------Limit--------		------Percent-------	
Lower (LSL)	0.900000	% < LSL	3.00000
Target	1.000000	% Between	92.00000
Upper (USL)	1.100000	% > USL	5.00000

Process Capability Indices

Index	Value	95% Confidence Limits	
Cp	0.526515	0.453237	0.599670
CPL	0.547575	0.446607	0.647299
CPU	0.505454	0.408856	0.600808
Cpk	0.505454	0.409407	0.601501
Cpm	0.525467	0.454973	0.601113

Example 8.2. Enhancing Reference Lines

See CAPSPEC1 in the SAS/QC Sample Library

A telecommunications company manufactures amplifiers to be used in telephones. Each amplifier is designed to boost the input signal by 5 decibels (dB). Since it is difficult to make every amplifier's boosting power exactly 5 decibels, the company decides that amplifiers that boost the input signal between 4 and 6 decibels are acceptable. Therefore, the target value is 5 decibels, and the lower and upper specifi-

cation limits are 4 and 6 decibels, respectively. The following data set contains the boosting powers of a sample of 75 amplifiers:

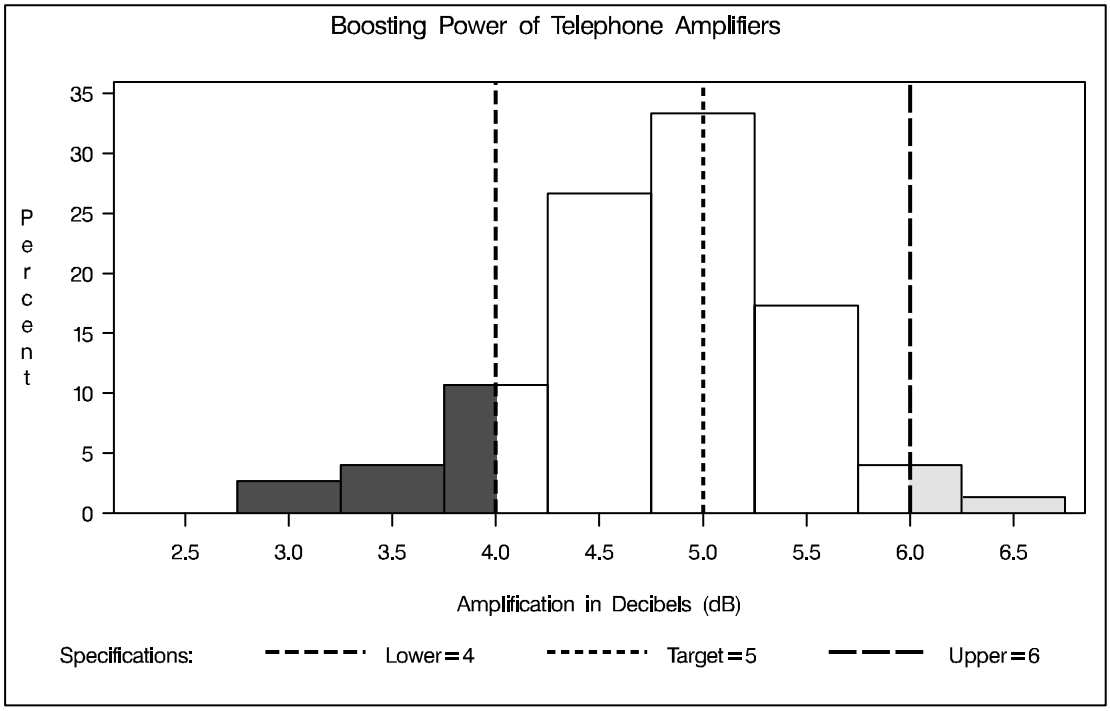
```
data amps;
   label decibels = 'Amplification in Decibels (dB)';
   input decibels @@;
datalines;
4.54 4.87 4.66 4.90 4.68 5.22 4.43 5.14 3.07 4.22
5.09 3.41 5.75 5.16 3.96 5.37 5.70 4.11 4.83 4.51
4.57 4.16 5.73 3.64 5.48 4.95 4.57 4.46 4.75 5.38
5.19 4.35 4.98 4.87 3.53 4.46 4.57 4.69 5.27 4.67
5.03 4.50 5.35 4.55 4.05 6.63 5.32 5.24 5.73 5.08
5.07 5.42 5.05 5.70 4.79 4.34 5.06 4.64 4.82 3.24
4.79 4.46 3.84 5.05 5.46 4.64 6.13 4.31 4.81 4.98
4.95 5.57 4.11 4.15 5.95
;
run;
```

The SPEC statement provides several options to control the appearance of reference lines for the specification limits and the target value. The following statements use the data set AMPS to create a histogram that demonstrates some of these options:

```
title 'Boosting Power of Telephone Amplifiers';
proc capability data=amps;
   spec  target = 5      lsl = 4         usl = 6
        ltarget = 2     llsl = 3        lusl = 4
        wtarget = 2     wlsl = 2        wusl = 2
        ctarget = black clsl = black    cusl = black
                        cleft = red    cright = yellow
                        pleft = solid  pright = solid;
   histogram decibels / cbarline = black;
run;
```

The resulting histogram is shown in Output 8.2.1. The LTARGET=, LLSL=, and LUSL= options control the line type of the reference lines for the target, lower specification limit, and upper specification limit, respectively. Likewise, the WTARGET=, WLSL=, and WUSL= options control the line widths. The CLEFT= and PLEFT= options control the color and pattern type used to fill the area to the left of the lower specification limit. Similarly, the CRIGHT= and PRIGHT= options control the color and pattern type used to fill the area to the right of the upper specification limit.

Output 8.2.1. Controlling the Appearance of Specification Limits



Example 8.3. Displaying a Confidence Interval for Cpk

See CAPSPEC3 in the SAS/QC Sample Library

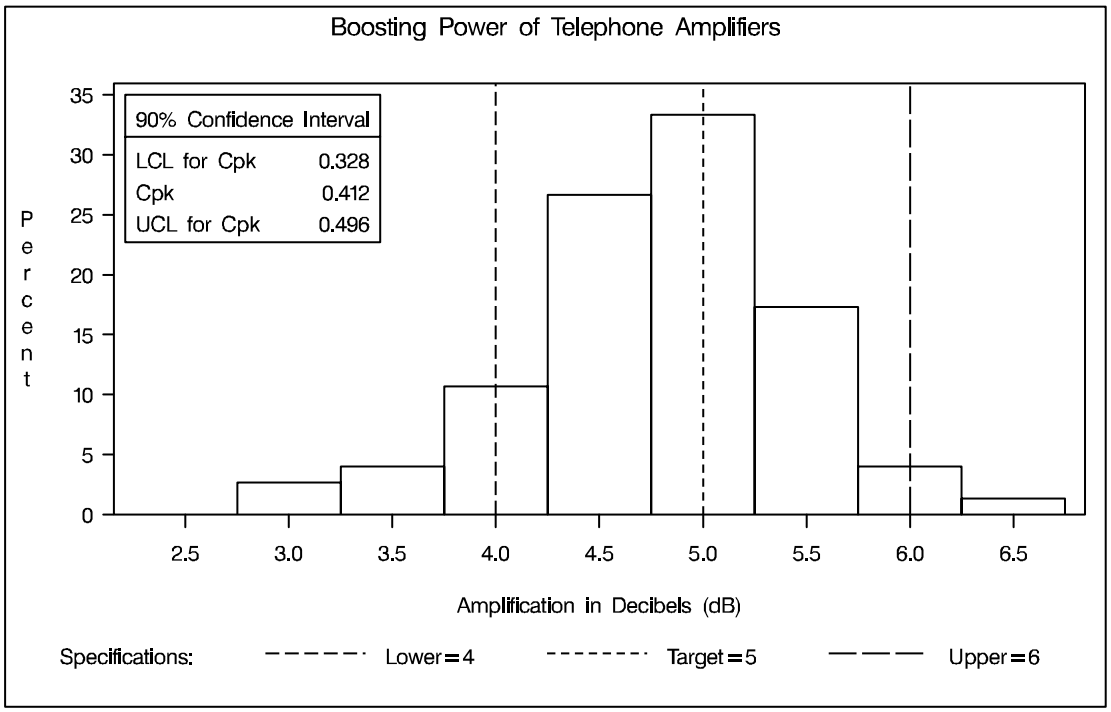
In this example, the capability index C_{pk} is computed for the amplification data in AMPS. To examine the accuracy of this estimate, the following statements calculate a 90% confidence interval for C_{pk}, then display the interval on a histogram (shown in Output 8.3.1) using the INSET statement:

```
title 'Boosting Power of Telephone Amplifiers';
proc capability data=amps noprint alpha=0.10;
   var decibels;
   spec  target = 5  lsl = 4  usl = 6
        ltarget = 2 llsl = 3 lusl = 4;
   histogram decibels;
   inset cpklcl cpk cpkucl / header = '90% Confidence Interval'
                             format = 6.3;
run;
```

The ALPHA= option in the PROC CAPABILITY statement controls the level of the confidence interval. In this case, the 90% confidence interval on C_{pk} is wide (from 0.328 to 0.496), indicating that the process may need adjustments in order to improve process variability. Confidence limits for capability indices can be displayed using the INSET statement (as shown in Output 8.3.1) or saved in an output data set using the OUTPUT statement. For formulas and details about capability indices,

see "Specialized Capability Indices" on page 208. For more information about the INSET statement, see Chapter 12, "INSET Statement," starting on page 353.

Output 8.3.1. Confidence Interval on C_{pk}



The following statements can be used to produce a table of process capability indices including the index C_{pk}:

```
ods select indices;
proc capability data=amps alpha=0.10;
   spec  target = 5  lsl = 4  usl = 6
        ltarget = 2 llsl = 3 lusl = 4;
   var decibels;
run;
```

Output 8.3.2. Process Capability Indices

The CAPABILITY Procedure
Variable: decibels (Amplification in Decibels (dB))

Process Capability Indices

Index	Value	90% Confidence Limits	
Cp	0.508962	0.439538	0.576922
CPL	0.411920	0.326620	0.495136
CPU	0.606004	0.501261	0.708127
Cpk	0.411920	0.327599	0.496241
Cpm	0.488674	0.425292	0.556732

Chapter 9
CDFPLOT Statement

Chapter Contents

OVERVIEW 227

GETTING STARTED 227
Creating a Cumulative Distribution Plot 227

SYNTAX 228
Summary of Options 229
Dictionary of Options 233

EXAMPLES 242
Example 9.1. Fitting a Normal Distribution 242
Example 9.2. Using Reference Lines with CDF Plots 243

Chapter 9
CDFPLOT Statement

Overview

The CDFPLOT statement plots the observed cumulative distribution function (*cdf*) of a variable, defined as

$$\begin{aligned} F_N(x) &= \text{percent of nonmissing values} \leq x \\ &= \frac{\text{number of values} \leq x}{N} \times 100\% \end{aligned}$$

where N is the number of nonmissing observations. The cdf is an increasing step function that has a vertical jump of $\frac{1}{N}$ at each value of x equal to an observed value. The cdf is also referred to as the empirical cumulative distribution function (ecdf).

You can use options in the CDFPLOT statement to

- superimpose specification limits
- superimpose fitted theoretical distributions (beta, exponential, gamma, lognormal, normal, and Weibull)
- specify graphical enhancements (such as color or text height)

Getting Started

Creating a Cumulative Distribution Plot

See CAPCDF1 in the SAS/QC Sample Library

This section introduces the CDFPLOT statement with a simple example. A company that produces fiber optic cord is interested in the breaking strength of the cord. The following statements create a data set named CORD, which contains 50 breaking strengths measured in pounds per square inch (psi), and they display the cdf plot in Figure 9.1. The plot shows a symmetric distribution with observations concentrated 6.9 and 7.1. The plot also shows that only a small percentage ($< 5\%$) of the observations are below the lower specification limit of 6.8.

```
data cord;
   label strength="Breaking Strength (psi)";
   input strength @@;
cards;
6.94 6.97 7.11 6.95 7.12 6.70 7.13 7.34 6.90 6.83
7.06 6.89 7.28 6.93 7.05 7.00 7.04 7.21 7.08 7.01
7.05 7.11 7.03 6.98 7.04 7.08 6.87 6.81 7.11 6.74
6.95 7.05 6.98 6.94 7.06 7.12 7.19 7.12 7.01 6.84
6.91 6.89 7.23 6.98 6.93 6.83 6.99 7.00 6.97 7.01
;
run;
```

```
legend2 FRAME CFRAME=ligr CBORDER=black POSITION=center;
title 'Cumulative Distribution Function of Breaking Strength';
proc capability data=cord noprint;
   spec lsl=6.8 llsl=2 clsl=black;
   cdf strength ;
run;
```

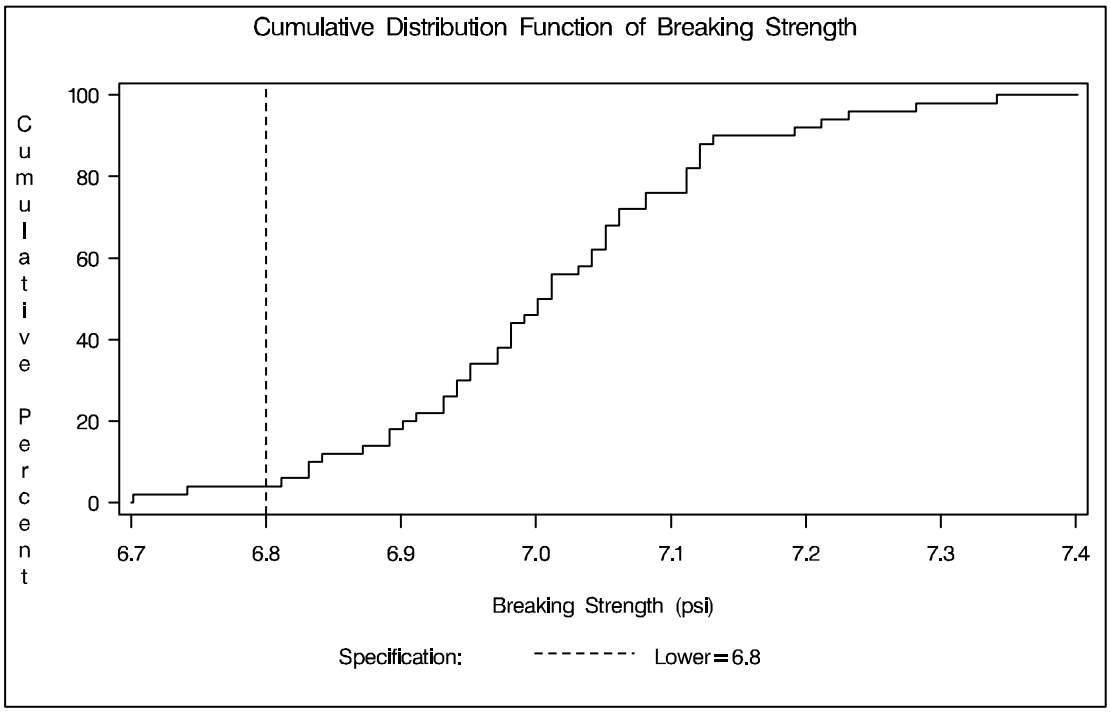


Figure 9.1. Cumulative Distribution Function

Syntax

The syntax for the CDFPLOT statement is as follows:

CDFPLOT *<variables>* **<** **/** *options* **>;**

You can specify the keyword CDF as an alias for CDFPLOT. You can specify any number of CDFPLOT statements after a PROC CAPABILITY statement. The components of the CDFPLOT statement are described as follows:

variables

specify variables for which to create cdf plots. If you specify a VAR statement, the *variables* must also be listed in the VAR statement. Otherwise, the *variables* can be any numeric variables in the input data set. If you do not specify *variables* in a CDFPLOT statement, then a cdf plot is created for each variable listed in the VAR statement, or for each numeric variable in the input data set if you do not use a VAR statement.

For example, suppose a data set named STEEL contains exactly three numeric variables, LENGTH, WIDTH and HEIGHT. The following statements create a cdf plot for each of the three variables:

```
proc capability data=steel;
   cdfplot;
run;
```

The following statements create a cdf plot for LENGTH and a cdf plot for WIDTH:

```
proc capability data=steel;
   var length width;
   cdfplot;
run;
```

The following statements create a cdf plot for WIDTH:

```
proc capability data=steel;
   var length width;
   cdfplot width;
run;
```

By default, the horizontal axis of a cdf plot is labeled with the *variable* name. If you specify a label for a variable, however, the label is used. The default vertical axis label is *Cumulative Percent*, and the axis is scaled in percent of observations.

If you specify a SPEC statement or a SPEC= data set in addition to the CDFPLOT statement, then the specification limits for each *variable* are displayed as reference lines and are identified in a legend.

options

add features to plots. All *options* appear after the slash (/) in the CDFPLOT statement. In the following example, the NORMAL option superimposes a normal cdf on the plot, and the CTEXT= option specifies the color of the text.

```
proc capability data=steel;
   cdfplot length / normal
                    ctext = yellow;
run;
```

Summary of Options

The following tables list all *options* by function. The "Dictionary of Options" on page 233 describes each option in detail.

Distribution Options

You can use the options listed in Table 9.1 to superimpose a fitted theoretical distribution function on your cdf plot.

Table 9.1. Main Distribution Options

BETA(*beta-options*)	plots two-parameter beta distribution function, parameters θ and σ assumed known
EXPONENTIAL(*exponential-options*)	plots one-parameter exponential distribution function, parameter θ assumed known
GAMMA(*gamma-options*)	plots two-parameter gamma distribution function, parameter θ assumed known
LOGNORMAL(*lognormal-options*)	plots two-parameter lognormal distribution function, parameter θ assumed known
NORMAL(*normal-options*)	plots normal distribution function
WEIBULL(*Weibull-options*)	plots two-parameter Weibull distribution function, parameter θ assumed known

You can specify options in parentheses after each distribution option to control features of the theoretical distribution function. For example, the following statements use the NORMAL option to superimpose a normal distribution:

```
proc capability;
   cdfplot / normal(mu=10 sigma=0.5 color=red);
run;
```

The COLOR= option specifies the color for the curve, and the *normal-options* MU= and SIGMA= specify the parameters $\mu = 10$ and $\sigma = 0.5$ for the distribution function. If you do not specify these parameters, maximum likelihood estimates are computed.

Table 9.2. Options Used with All Distribution Options

COLOR=*color*	specifies color of theoretical distribution function
L=*linetype*	specifies line type of theoretical distribution function
SYMBOL='*character*'	specifies *character* used to plot theoretical distribution function if cdf plot is produced on a line printer
W=n	specifies width of theoretical distribution function

Table 9.3. Beta-Options

ALPHA=*value*	specifies first shape parameter α for beta distribution function
BETA=*value*	specifies second shape parameter β for beta distribution function
SIGMA=*value*	specifies scale parameter σ for beta distribution function
THETA=*value*	specifies lower threshold parameter θ for beta distribution function

Table 9.4. Exponential-Options

SIGMA=*value*	specifies scale parameter σ for exponential distribution function
THETA=*value*	specifies threshold parameter θ for exponential distribution function

Table 9.5. Gamma-Options

ALPHADELTA=*value*	specifies change in successive estimates of α at which the Newton-Raphson approximation of $\hat{\alpha}$ terminates
ALPHAINITIAL=*value*	specifies initial value for α in the Newton-Raphson approximation of $\hat{\alpha}$
MAXITER=n	specifies maximum number of iterations in the Newton-Raphson approximation of $\hat{\alpha}$
SIGMA=*value*	specifies scale parameter σ for gamma distribution function
ALPHA=*value*	specifies shape parameter α for gamma distribution function
THETA=*value*	specifies threshold parameter θ for gamma distribution function

Table 9.6. Lognormal-Options

ZETA=*value*	specifies scale parameter ζ for lognormal distribution function
SIGMA=*value*	specifies shape parameter σ for lognormal distribution function
THETA=*value*	specifies threshold parameter θ for lognormal distribution function

Table 9.7. Normal-Options

MU=*value*	specifies mean μ for normal distribution function
SIGMA=*value*	specifies standard deviation σ for normal distribution function

Table 9.8. Weibull-Options

C=*value*	specifies shape parameter c for Weibull distribution function
CDELTA=*value*	specifies change in successive estimates of c at which the Newton-Raphson approximation of $\hat{c}$ terminates
CINITIAL=*value*	specifies initial value for c in the Newton-Raphson approximation of $\hat{c}$
MAXITER=*value*	specifies maximum number of iterations in the Newton-Raphson approximation of $\hat{c}$
SIGMA=*value*	specifies scale parameter σ for Weibull distribution function
THETA=*value*	specifies threshold parameter θ for Weibull distribution function

General Options

Table 9.9. Options to Enhance Plots Produced on Graphics Devices

ANNOTATE=*SAS-data-set*	specifies annotate data set
CAXIS=*color*	specifies color for axis
CFRAME=*color*	specifies color for frame
CHREF=*color*	specifies color for HREF= lines
CTEXT=*color*	specifies color for text
CVREF=*color*	specifies color for VREF= lines
DESCRIPTION='*string*'	specifies description for graphics catalog member
FONT=*font*	specifies software font for text
HAXIS=*name*	specifies AXIS statement for horizontal axis
HMINOR=n	specifies number of horizontal minor tick marks
LEGEND=*name* | NONE	identifies LEGEND statement
LHREF=*linetype*	specifies line style for HREF= lines
LVREF=*linetype*	specifies line style for VREF= lines
NAME='*string*'	specifies name for plot in graphics catalog
VAXIS=*name*	specifies AXIS statement for vertical axis
VMINOR=n	specifies number of vertical minor tick marks

Table 9.10. Options to Enhance Plots Produced on Line Printers

CDFSYMBOL='*character*'	specifies character for plotted points
HREFCHAR='*character*'	specifies line character for HREF= lines
VREFCHAR='*character*'	specifies line character for VREF= lines

Table 9.11. General Plot Layout Options

HREF=*value-list*	specifies reference lines perpendicular to the horizontal axis
HREFLABELS='*label1*'...'*labeln*'	specifies labels for HREF= lines
NOCDFLEGEND	suppresses legend for superimposed theoretical cdf
NOECDF	suppresses plot of empirical (observed) distribution function
NOFRAME	suppresses frame around plotting area
NOLEGEND	suppresses legend
NOSPECLEGEND	suppresses specifications legend
VREF=*value-list*	specifies reference lines perpendicular to the vertical axis
VREFLABELS='*label1*'...'*labeln*'	specifies labels for VREF= lines
VSCALE=PERCENT | PROPORTION	specifies scale for vertical axis

Dictionary of Options

The following entries provide detailed descriptions of the *options* in the CDFPLOT statement. The marginal notes *Graphics* and *Line Printer* identify options that can be used only with graphics devices and line printers, respectively.

ALPHA=*value*

specifies the shape parameter α for distribution functions requested with the BETA and GAMMA options. Enclose the ALPHA= option in parentheses after the BETA or GAMMA keywords. If you do not specify a value for α, the procedure calculates a maximum likelihood estimate. For examples, see the entries for the BETA and GAMMA options.

ALPHADELTA=*value*

specifies the change in successive estimates of $\hat{\alpha}$ at which iteration terminates in the Newton-Raphson approximation of the maximum likelihood estimate of α for curves requested by the GAMMA option. Enclose the ALPHADELTA= option in parentheses after the GAMMA keyword. Iteration continues until the change in α is less than the value specified or the number of iterations exceeds the value of the MAXITER= option (see page 238). The default value is 0.00001.

ALPHAINITIAL=*value*

specifies the initial value for $\hat{\alpha}$ in the Newton-Raphson approximation of the maximum likelihood estimate of α for fitted gamma distributions requested with the GAMMA option. Enclose the ALPHAINITIAL= option in parentheses after the GAMMA keyword. The default value is Thom's approximation of the estimate of α (refer to Johnson *et al.* (1995).

ANNOTATE=*SAS-data-set*
ANNO=*SAS-data-set*

Graphics

specifies an annotate data set, as described in *SAS/GRAPH Software: Reference*, that allows you to add features to the cdf plot. The ANNOTATE= data set you specify in the CDFPLOT statement is used for all plots created by the statement. You can also specify an ANNOTATE= data set in the PROC CAPABILITY statement, which provides annotate information used for all plots created by the procedure (see "ANNOTATE= Data Sets" on page 189).

BETA<**(***beta-options* **)**>

displays a fitted beta distribution function on the cdf plot. The equation of the fitted cdf is

$$F(x) = \begin{cases} 0 & \text{for } x \leq \theta \\ I_{\frac{x-\theta}{\sigma}}(\alpha, \beta) & \text{for } \theta < x < \theta + \sigma \\ 1 & \text{for } x \geq \sigma + \theta \end{cases}$$

where $I_y(\alpha, \beta)$ is the incomplete beta function, and

$\theta =$ lower threshold parameter (lower endpoint)
$\sigma =$ scale parameter $(\sigma > 0)$
$\alpha =$ shape parameter $(\alpha > 0)$
$\beta =$ shape parameter $(\beta > 0)$

The beta distribution is bounded below by the parameter θ and above by the value $\theta + \sigma$. You can specify θ and σ using the THETA= and SIGMA= *beta-options*, as illustrated in the following statements, which fit a beta distribution bounded between 50 and 75. The default values for θ and σ are 0 and 1, respectively.

```
proc capability;
   cdfplot / beta(theta=50 sigma=25);
run;
```

The beta distribution has two shape parameters, α and β. If these parameters are known, you can specify their values with the ALPHA= and BETA= *beta-options*. If you do not specify values for α and β, the procedure calculates maximum likelihood estimates.

The BETA option can appear only once in a CDFPLOT statement. Table 9.2 on page 230 and Table 9.3 on page 230 list options you can specify with the BETA distribution option.

BETA=*value*
B=*value*
specifies the second shape parameter β for beta distribution functions requested by the BETA option. Enclose the BETA= option in parentheses after the BETA keyword. If you do not specify a value for β, the procedure calculates a maximum likelihood estimate. For examples, see the preceding entry for the BETA option.

C=*value*
specifies the shape parameter c for Weibull distribution functions requested with the WEIBULL option. Enclose the C= option in parentheses after the WEIBULL keyword. If you do not specify a value for c, the procedure calculates a maximum likelihood estimate. You can specify the SHAPE= option as an alias for the C= option.

CAXIS=*color*
CAXES=*color*
Graphics
specifies the color used for the axes and tick marks. This option overrides any COLOR= specifications in an AXIS statement. The default is the first color in the device color list.

CDELTA=*value*
specifies the change in successive estimates of c at which iterations terminate in the Newton-Raphson approximation of the maximum likelihood estimate of c for fitted Weibull curves requested by the WEIBULL option. Enclose the CDELTA= option in parentheses after the WEIBULL keyword. Iteration continues until the change in c between consecutive steps is less than the *value* specified or until the number of iterations exceeds the value of the MAXITER= option (see page 238). The default value is 0.00001.

CDFSYMBOL='*character***'**
Line Printer
specifies the character used to plot the points when the cdf plot is produced on a line printer. The default is the plus sign (+). Use the SYMBOL statement to control the plotting symbol when the plot is produced on a graphics device.

CFRAME=*color*
CFR=*color*
specifies the color for the area enclosed by the axes and frame. This area is not shaded by default. *Graphics*

CHREF=*color*
CH=*color*
specifies the color for lines requested by the HREF= option. The default is the first color in the device color list. *Graphics*

CINITIAL=*value*
specifies the initial value for $\hat{c}$ in the Newton-Raphson approximation of the maximum likelihood estimate of c for Weibull distributions requested by the WEIBULL option. The default value is 1.8 (refer to Johnson *et al.* 1995).

COLOR=*color*
specifies the color of the fitted distribution curve. Enclose the COLOR= option in parentheses after a distribution option. For a syntax example, see page 230. *Graphics*

CTEXT=*color*
specifies the color for tick mark values and axis labels. The default is the color specified for the CTEXT= option in the most recent GOPTIONS statement. *Graphics*

CVREF=*color*
CV=*color*
specifies the color for lines requested by the VREF= option. The default is the first color in the device color list. *Graphics*

DESCRIPTION='*string***'**
DES='*string***'**
specifies a description, up to 40 characters, that appears in the PROC GREPLAY master menu. The default is the variable name. *Graphics*

EXPONENTIAL<(*exponential-options* **)>**
EXP<(*exponential-options* **)>**
displays a fitted exponential distribution function on the cdf plot. The equation of the fitted cdf is

$$F(x) = \begin{cases} 0 & \text{for } x \leq \theta \\ 1 - \exp\left(-\frac{x-\theta}{\sigma}\right) & \text{for } x > \theta \end{cases}$$

where

θ = threshold parameter

σ = scale parameter $(\sigma > 0)$ The parameter θ must be less than or equal to the minimum data value. You can specify θ with the THETA= *exponential-option*. The default value for θ is 0. You can specify σ with the SIGMA= *exponential-option*. By default, a maximum likelihood estimate is computed for σ. For example, the following statements fit an exponential distribution with $\theta = 10$ and a maximum likelihood estimate for σ:

```
proc capability;
   cdfplot / exponential(theta=10 l=2 color=green);
run;
```

The exponential curve is green and has a line type of 2.

The EXPONENTIAL option can appear only once in a CDFPLOT statement. Table 9.2 on page 230 and Table 9.4 on page 231 list the options you can specify with the EXPONENTIAL option.

FONT=*font*

Graphics

specifies a software font for reference line and axis labels. You can also specify fonts for axis labels in an AXIS statement. The FONT= font takes precedence over the FTEXT= font specified in the most recent GOPTIONS statement. Hardware characters are used by default.

GAMMA<**(***gamma-options***)**>

displays a fitted gamma distribution function on the cdf plot. The equation of the fitted cdf is

$$F(x) = \begin{cases} 0 & \text{for } x \leq \theta \\ \frac{1}{\Gamma(\alpha)\sigma} \int_{\theta}^{x} \left(\frac{t-\theta}{\sigma}\right)^{\alpha-1} \exp\left(-\frac{t-\theta}{\sigma}\right) dt & \text{for } x > \theta \end{cases}$$

where

θ = threshold parameter
σ = scale parameter $(\sigma > 0)$
α = shape parameter $(\alpha > 0)$ The parameter θ for the gamma distribution must be less than the minimum data value. You can specify θ with the THETA= *gamma-option*. The default value for θ is 0. In addition, the gamma distribution has a shape parameter α and a scale parameter σ. You can specify these parameters with the ALPHA= and SIGMA= *gamma-options*. By default, maximum likelihood estimates are computed for α and σ. For example, the following statements fit a gamma distribution function with $\theta = 4$ and maximum likelihood estimates for α and σ:

```
proc capability;
   cdfplot / gamma(theta=4);
run;
```

Note that the maximum likelihood estimate of α is calculated iteratively using the Newton-Raphson approximation. The *gamma-options* ALPHADELTA=, ALPHAINITIAL=, and MAXITER= control the approximation.

The GAMMA option can appear only once in a CDFPLOT statement. Table 9.2 on page 230 and Table 9.5 on page 231 list the options you can specify with the GAMMA option.

HAXIS=*name*

Graphics

specifies the name of an AXIS statement describing the horizontal axis.

HMINOR=*n*
HM=*n*

specifies the number of minor tick marks between each major tick mark on the horizontal axis. Minor tick marks are not labeled. The default is 0. *Graphics*

HREF=*value-list*

draws reference lines perpendicular to the horizontal axis at the values specified. See Output 9.2.1 on page 244 for an example that uses the similar VREF= option. See also the entries for the CHREF=, HREFCHAR=, and LHREF= options.

HREFCHAR=*'character'*

specifies the character used to form the lines requested by the HREF= option. The default is the vertical bar (|). *Line Printer*

HREFLABELS=*'label1' . . . 'labeln'*
HREFLABEL=*'label1' . . . 'labeln'*
HREFLAB=*'label1' . . . 'labeln'*

specifies labels for the lines requested by the HREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters. See Output 9.2.1 on page 244 for an example that uses the similar VREFLABELS= option.

LEGEND=*name* **| NONE**

specifies the name of a LEGEND statement describing the legend for specification limit reference lines and superimposed distribution functions. Specifying LEGEND=NONE, which suppresses all legend information, is equivalent to specifying the NOLEGEND option. *Graphics*

LHREF=*linetype*
LH=*linetype*

specifies the line type for lines requested by the HREF= option. The default is 2, which produces a dashed line. *Graphics*

LOGNORMAL<**(***lognormal-options***)**>

displays a fitted lognormal distribution function on the cdf plot. The equation of the fitted cdf is

$$F(x) = \begin{cases} 0 & \text{for } x \le \theta \\ \Phi\left(\frac{\log(x-\theta)-\zeta}{\sigma}\right) & \text{for } x > \theta \end{cases}$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function, and

θ = threshold parameter
ζ = scale parameter
σ = shape parameter ($\sigma > 0$) The parameter θ for the lognormal distribution must be less than the minimum data value. You can specify θ with the THETA= *lognormal-option*. The default value for θ is 0. In addition, the lognormal distribution has a shape parameter σ and a scale parameter ζ. You can specify these parameters with the SIGMA= and ZETA= *lognormal-options*. By default, maximum likelihood estimates are computed for σ and ζ. For example, the following statements fit a

lognormal distribution function with $\theta = 10$ and maximum likelihood estimates for σ and ζ:

```
proc capability;
   cdfplot / lognormal(theta = 10);
run;
```

The LOGNORMAL option can appear only once in a CDFPLOT statement. Table 9.2 on page 230 and Table 9.6 on page 231 list options that you can specify with the LOGNORMAL option.

LVREF=*linetype*
LV=*linetype*

Graphics

specifies the line type for lines requested by the VREF= option. The default is 2, which produces a dashed line.

MAXITER=*n*

specifies the maximum number of iterations in the Newton-Raphson approximation of the maximum likelihood estimate of α for fitted gamma distributions requested with the GAMMA option and c for fitted Weibull distributions requested with the WEIBULL option. Enclose the MAXITER= option in parentheses after the GAMMA or WEIBULL keywords. The default value of n is 20.

MU=*value*

specifies the parameter μ for normal distribution functions requested with the NORMAL option. Enclose the MU= option in parentheses after the NORMAL keyword. The default value is the sample mean. For an example, see the entry for the NORMAL option.

NAME=*'string'*

Graphics

specifies a name for the plot, up to eight characters, that appears in the PROC GREPLAY master menu. The default is 'CAPABILI'.

NOCDFLEGEND

suppresses the legend for the superimposed theoretical cumulative distribution function.

NOECDF

suppresses the observed distribution function (the empirical cumulative distribution function) of the variable, which is drawn by default. This option allows you to create theoretical cdf plots without displaying the data distribution. The NOECDF option can be used only with a theoretical distribution (such as the NORMAL option).

NOFRAME

suppresses the frame around the subplot area.

NOLEGEND

suppresses legends for specification limits, theoretical distribution functions, and hidden observations. Specifying the NOLEGEND option is equivalent to specifying LEGEND=NONE.

NORMAL<**(***normal-options***)**>
displays a fitted normal distribution function on the cdf plot. The equation of the fitted cdf is

$$F(x) = \Phi\left(\frac{x-\mu}{\sigma}\right) \quad \text{for } -\infty < x < \infty$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function, and

μ = mean
σ = standard deviation ($\sigma > 0$)

You can specify known values for μ and σ with the MU= and SIGMA= *normal-options*, as shown in the following statements:

```
proc capability;
   cdfplot / normal(mu=14 sigma=.05);
run;
```

By default, the sample mean and sample standard deviation are calculated for μ and σ. The NORMAL option can appear only once in a CDFPLOT statement. For an example, see Output 9.1.1 on page 242. Table 9.2 on page 230 and Table 9.7 on page 231 list options that you can specify with the NORMAL option.

NOSPECLEGEND
NOSPECL
suppresses the portion of the legend for specification limit reference lines.

SCALE=*value*
is an alias for the SIGMA= option for distributions requested by the BETA, EXPONENTIAL, GAMMA, and WEIBULL options and for the ZETA= option for distributions requested by the LOGNORMAL option.

SHAPE=*value*
is an alias for the ALPHA= option for distributions requested by the GAMMA option, for the SIGMA= option for distributions requested by the LOGNORMAL option, and for the C= option for distributions requested by the WEIBULL option.

SIGMA=*value*
specifies the parameter σ for distribution functions requested by the BETA, EXPONENTIAL, GAMMA, LOGNORMAL, NORMAL, and WEIBULL options. Enclose the SIGMA= option in parentheses after the distribution keyword. The following table summarizes the use of the SIGMA= option:

Distribution Option	SIGMA= Specifies	Default Value	Alias
BETA	scale parameter σ	1	SCALE=
EXPONENTIAL GAMMA WEIBULL	scale parameter σ	maximum likelihood estimate	SCALE=
LOGNORMAL	shape parameter σ	maximum likelihood estimate	SHAPE=
NORMAL	scale parameter σ	standard deviation	

SYMBOL='*character***'**

Line Printer

specifies the *character* used to plot the theoretical distribution function if the cdf plot is produced on a line printer. Enclose the SYMBOL= option in parentheses after the distribution option. The default character is the first letter of the distribution option keyword.

THETA=*value*

specifies the lower threshold parameter θ for theoretical cumulative distribution functions requested with the BETA, EXPONENTIAL, GAMMA, LOGNORMAL, and WEIBULL options. Enclose the THETA= option in parentheses after the distribution keyword. The default *value* is 0.

THRESHOLD=*value*

is an alias for the THETA= option. See the preceding entry for the THETA= option.

VAXIS=*name*

Graphics

specifies the name of an AXIS statement describing the vertical axis. See Output 9.1.1 on page 242 for an example.

VMINOR=*n*
VM=*n*

Graphics

specifies the number of minor tick marks between each major tick mark on the vertical axis. Minor tick marks are not labeled. The default is 0.

VREF=*value-list*

draws reference lines perpendicular to the vertical axis at the values specified. See Output 9.2.1 on page 244 for an example. See also the entries for the CVREF=, LVREF=, and VREFCHAR= options.

VREFCHAR='*character***'**

Line Printer

specifies the character used to form the lines requested by the VREF= option for a line printer. The default is the hyphen (-).

VREFLABELS=*'label1' . . . 'labeln'*
VREFLABEL=*'label1' . . . 'labeln'*
VREFLAB=*'label1' . . . 'labeln'*

specifies labels for the lines requested by the VREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters. See Output 9.2.1 on page 244 for an example.

VSCALE=PERCENT | PROPORTION

specifies the scale of the vertical axis. The value PERCENT scales the data in units of percent of observations per data unit. The value PROPORTION scales the data in units of proportion of observations per data unit. The default is PERCENT.

W=*n*

Graphics

specifies the width in pixels of the superimposed theoretical distribution. Enclose the W= option in parentheses after the distribution option. For example, the following statements display an exponential distribution with a width of 3. The default is 1.

```
proc capability;
   cdfplot / exponential(w=3);
run;
```

WEIBULL<**(***Weibull-options***)**>

displays a fitted Weibull distribution function on the cdf plot. The equation of the fitted cdf is

$$F(x) = \begin{cases} 0 & \text{for } x \leq \theta \\ 1 - \exp\left(-\left(\frac{x-\theta}{\sigma}\right)^c\right) & \text{for } x > \theta \end{cases}$$

where

θ = threshold parameter
σ = scale parameter $(\sigma > 0)$
c = shape parameter $(c > 0)$

The parameter θ must be less than the minimum data value. You can specify θ with the THETA= *Weibull-option*. The default value for θ is 0. In addition, the Weibull distribution has a shape parameter c and a scale parameter σ. You can specify these parameters with the SIGMA= and C= *Weibull-options*. By default, maximum likelihood estimates are computed for c and σ. For example, the following statements fit a Weibull distribution function with $\theta = 15$ and maximum likelihood estimates for σ and c:

```
proc capability;
   cdfplot / weibull(theta=15);
run;
```

Note that the maximum likelihood estimate of c is calculated iteratively using the Newton-Raphson approximation. The *Weibull-options* CDELTA=, CINITIAL=, and MAXITER= control the approximation.

The WEIBULL option can appear only once in a CDFPLOT statement. Table 9.2 on page 230 and Table 9.8 on page 231 list options that you can specify with the WEIBULL option.

ZETA=*value*

specifies a value for the scale parameter ζ for a lognormal distribution function requested with the LOGNORMAL option. Enclose the ZETA= option in parentheses after the LOGNORMAL keyword. If you do not specify a *value* for ζ, a maximum likelihood estimate is computed. You can specify the SCALE= option as an alias for the ZETA= option.

Examples

This section illustrates how to display a fitted distribution function, inset tables, and display reference lines on your cdf plot.

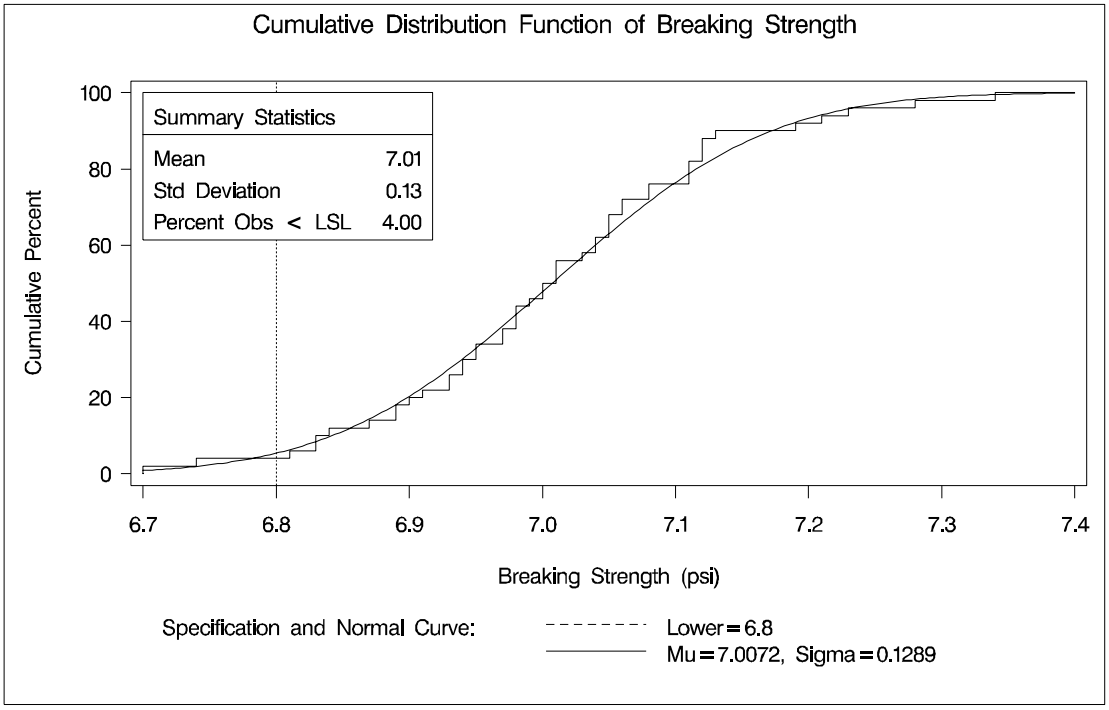
Example 9.1. Fitting a Normal Distribution

See CAPCDF1 in the SAS/QC Sample Library

You can use the CDFPLOT statement to fit any of six theoretical distributions (beta, exponential, gamma, lognormal, normal, and Weibull) and superimpose them on the cdf plot. The following statements use the NORMAL option to display a fitted normal distribution function on a cdf plot of breaking strengths. The data set CORD is given in Figure 9.1 on page 228, and the plot is shown in Output 9.1.1.

```
title 'Cumulative Distribution Function of Breaking Strength';
proc capability data=cord noprint;
   spec lsl=6.8 llsl=2 clsl=black;
   cdf strength / normal(color=black)
                  vaxis = axis1;
   inset mean std pctlss / cfill  = blank
                           format = 5.2
                           header = "Summary Statistics";
   axis1 label=(a=90 r=0);
run;
```

Output 9.1.1. Superimposed Normal Distribution Function



The NORMAL option requests the fitted curve. The VAXIS= option specifies the AXIS statement controlling the vertical axis. The AXIS1 statement is used to rotate the vertical axis label *Cumulative Percent*. The INSET statement requests an inset containing the mean, the standard deviation, and the percent of observations below the lower specification limit. For more information about the INSET statement, see Chapter 12, "INSET Statement," starting on page 353. The SPEC statement requests a lower specification limit at 6.8 with a line type of 2 (a dashed line). For more information about the SPEC statement, see "Syntax for the SPEC Statement" on page 183.

The agreement between the empirical and the normal distribution functions in Output 9.1.1 is evidence that the normal distribution is an appropriate model for the distribution of breaking strengths.

The CAPABILITY procedure provides a variety of other tools for assessing goodness of fit. Goodness-of-fit tests (see "Printed Output" on page 321) provide a quantitative assessment of a proposed distribution. Probability and Q-Q plots, created with the PROBPLOT (page 429), QQPLOT (page 461), and PPPLOT (page 407) statements, provide effective graphical diagnostics.

Example 9.2. Using Reference Lines with CDF Plots

See CAPCDF1 in the SAS/QC Sample Library

Customer requirements dictate that the breaking strengths in the previous example have upper and lower specification limits of 7.2 and 6.8 psi, respectively. Moreover, less than 5% of the cords can have breaking strengths outside the limits.

The following statements create a cdf plot with reference lines at the 5% and 95% cumulative percent levels:

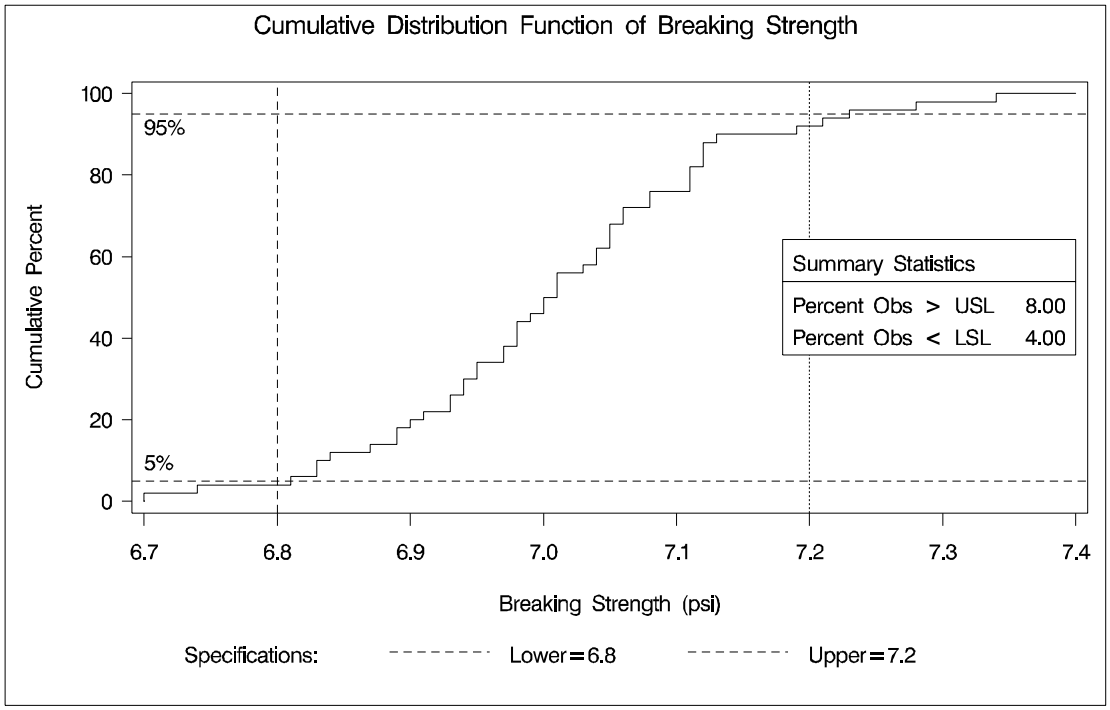
```
title 'Cumulative Distribution Function of Breaking Strength';
proc capability data=cord noprint;
   spec lsl=6.8 llsl=2 usl=7.2 lusl=2 cusl=black clsl=black;
   cdf strength /
                  vref         = 5 95
                  cvref        = black
                  vreflabels   = '5%' '95%'
                  vaxis        = axis1;
   inset pctgtr pctlss / cfill  = blank
                         format = 5.2
                         pos    = e
                         header = "Summary Statistics";
   axis1 label=(a=90 r=0);
run;
```

The INSET statement requests an inset with the percentages of measurements above the upper limit and below the lower limit. For more information about the INSET statement, see Chapter 12, "INSET Statement," starting on page 353.

In Output 9.2.1, the empirical cdf is below the intersection between the lower specification limit line and the 5% line, so less than 5% of the measurements are below the lower limit. The ecdf, however, is *also* below the intersection between the upper

specification limit line and the 95% line, implying that *more* than 5% of the measurements are greater than the upper limit. Thus, the goal of having less than 5% of the measurements above the upper specification limit has not been met.

Output 9.2.1. Reference Lines with a Cumulative Distribution Function Plot



Chapter 10
COMPHISTOGRAM Statement

Chapter Contents

OVERVIEW 247

GETTING STARTED 247
Creating a One-Way Comparative Histogram 248
Adding Fitted Normal Curves to a Comparative Histogram 249

SYNTAX 251
Summary of Options 253
Dictionary of Options 257

EXAMPLES 272
Example 10.1. Adding Insets with Descriptive Statistics 272
Example 10.2. Creating a Two-Way Comparative Histogram 274

Chapter 10
COMPHISTOGRAM Statement

Overview

Comparative histograms are useful for comparing the distribution of a process variable across levels of classification variables. You can use the COMPHISTOGRAM statement to create one-way and two-way comparative histograms. When used with a single classification variable, the COMPHISTOGRAM statement displays an array of component histograms (stacked or side-by-side), one for each level of the classification variable. When used with two classification variables, the COMPHISTOGRAM statement displays a matrix of component histograms, one for each combination of levels of the classification variables.

In quality improvement applications, typical uses of comparative histograms include

- comparing the capability of a process before and after an improvement
- comparing process capabilities of two or more suppliers
- exploring stratification in process data due to different lots, machines, manufacturing methods, and so forth
- studying the evolution of process capability over successive time periods

You can use options in the COMPHISTOGRAM statement to

- specify the midpoints or endpoints for histogram intervals
- specify the number of rows and/or columns of component histograms
- display specification limits on the component histograms
- display density curves for fitted normal distributions
- display kernel density estimates
- request graphical enhancements
- inset summary statistics and process capability indices on the component histograms

Getting Started

This section introduces the COMPHISTOGRAM statement with examples that illustrate commonly used options. Complete syntax for the COMPHISTOGRAM statement is presented in the "Syntax" section on page 251, and advanced examples are given in the "Examples" section on page 272.

Creating a One-Way Comparative Histogram

See CAPCMH1 in the SAS/QC Sample Library

The effective channel length (in microns) is measured for 1225 field effect transistors. The channel lengths are saved as values of the variable LENGTH in a SAS data set named CHANNEL:

```
data channel;
   length lot $ 16;
   input length @@;
   select;
        when (_n_ <= 425) Lot='Lot 1';
        when (_n_ >= 926) Lot='Lot 3';
        otherwise Lot='Lot 2';
     end;
   datalines;
0.91 1.01 0.95 1.13 1.12 0.86 0.96 1.17 1.36 1.10
0.98 1.27 1.13 0.92 1.15 1.26 1.14 0.88 1.03 1.00
0.98 0.94 1.09 0.92 1.10 0.95 1.05 1.05 1.11 1.15
1.11 0.98 0.78 1.09 0.94 1.05 0.89 1.16 0.88 1.19
1.01 1.08 1.19 0.94 0.92 1.27 0.90 0.88 1.38 1.02

  ...

1.80 2.35 2.23 1.96 2.16 2.08 2.06 2.03 2.18 1.83
2.13 2.05 1.90 2.07 2.15 1.96 2.15 1.89 2.15 2.04
1.95 1.93 2.22 1.74 1.91
;
```

The data set CHANNEL is also used in Example 11.5 on page 344, where a kernel density estimate is superimposed on the histogram of channel lengths. The display in Output 11.5.1 on page 346 reveals that there are three distinct peaks in the process distribution. To investigate whether these peaks (modes) in the histogram are related to the lot source, you can create a comparative histogram using LOT as a classification variable. The following statements create the comparative histogram shown in Figure 10.1:

```
goptions lfactor = 3
         htext   = 2.5 pct
         htitle  = 3.0 pct;
title "Comparative Analysis of Lot Source";
proc capability data=channel noprint;
   specs lsl = 0.8  clsl = black  llsl = 2
         usl = 2.0  cusl = black  lusl = 3 ;
   comphist length / class      = lot
                     nrows      = 3
                     nlegend    = 'Lot Size'
                     nlegendpos = nw;
   label lot = 'Transistor Source';
run;
```

The COMPHISTOGRAM statement requests a comparative histogram for the process variable LENGTH. The CLASS= option requests a component histogram for each level (distinct value) of the classification variable LOT. The option NROWS=3 stacks the histograms three to a page. The NLEGEND= option adds a sample size legend to each component histogram, and the option NLEGENDPOS=NW positions each legend in the northwest corner. The SPEC statement provides the specification limits displayed as vertical reference lines. See "Dictionary of Options" on page 257 for descriptions of these options, and see "Syntax for the SPEC Statement" on page 183 for details of the SPEC statement.

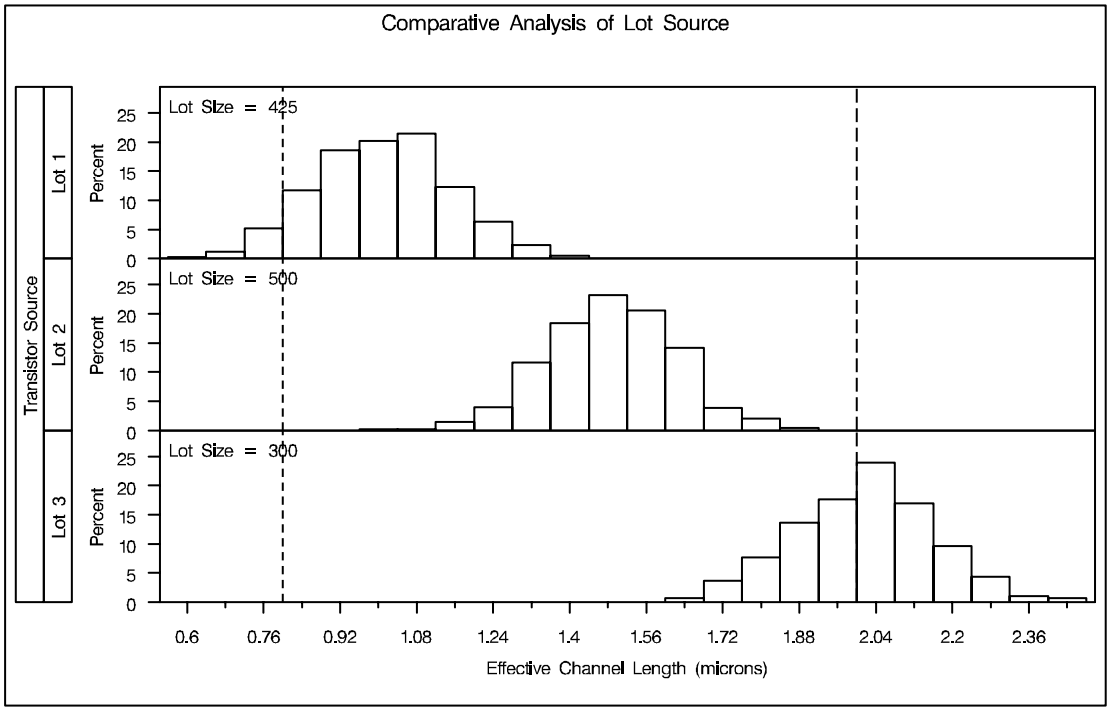


Figure 10.1. Comparison by Lot Source

Adding Fitted Normal Curves to a Comparative Histogram

See CAPCMH1 in the SAS/QC Sample Library

In Figure 10.1, it appears that each lot produces transistors with channel lengths that are normally distributed. The following statements use the NORMAL option to fit a normal distribution to the data for each lot (the observations corresponding to a specific level of the classification variable are referred to as a *cell*). The normal parameters μ and σ are estimated from the data for each lot, and the curves are superimposed on each component histogram.

```
goptions lfactor = 3
         htext   = 2.5 pct
         htitle  = 3.0 pct;
title "Comparative Analysis of Lot Source";
proc capability data=channel noprint;
```

```
   specs lsl = 0.8  clsl = black  llsl = 2
         usl = 2.0  cusl = black  lusl = 3 ;
   comphist length / class     = lot
                     nrows     = 3
                     intertile = 1
                     cprop     = orange
                     normal(color = black);
   label lot = 'Transistor Source';
run;
```

The comparative histogram is displayed in Figure 10.2.

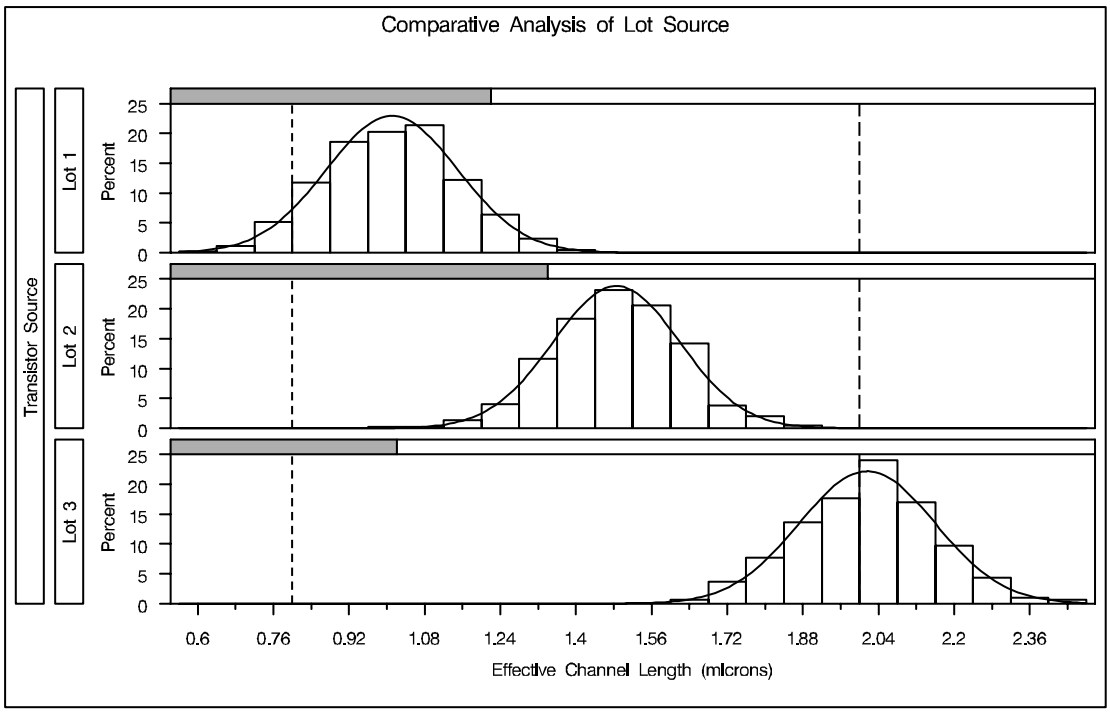


Figure 10.2. Fitting Normal Curves

Specifying INTERTILE=1 inserts a space of one percent screen unit between the framed areas, which are referred to as *tiles*. The shaded bars, added with the CPROP= option, represent the relative frequency of observations in each cell. See "Dictionary of Options" on page 257 for details concerning these options.

Syntax

The syntax for the COMPHISTOGRAM statement is as follows:

COMPHISTOGRAM <*variables* > **/ CLASS=(***class-variables***)** <*options* >**;**

You can specify the keyword COMPHIST as an alias for COMPHISTOGRAM. You can use any number of COMPHISTOGRAM statements after a PROC CAPABILITY statement.

To create a comparative histogram, you must specify at least one *variable* and either one or two *class-variables* (also referred to as *classification variables*). * The COMPHISTOGRAM statement displays a component histogram for each level of the *class-variables* using the values of the *variable*. The observations in a given level are referred to as a *cell*.

The components of the COMPHISTOGRAM statement are described as follows:

variables
are the process variables for which comparative histograms are to be created. If you specify a VAR statement, the *variables* must also be listed in the VAR statement. Otherwise, *variables* can be any numeric variables in the input data set that are not also listed as *class-variables*. If you do not specify *variables* in a COMPHISTOGRAM statement or a VAR statement, then by default a comparative histogram is created for each numeric variable in the DATA= data set that is not used as a *class-variable*. If you use a VAR statement and do not specify *variables* in the COMPHISTOGRAM statement, then by default a comparative histogram is created for each variable listed in the VAR statement.

For example, suppose a data set named STEEL contains two process variables named LENGTH and WIDTH, a numeric classification variable named LOT, and a character classification variable named DAY. The following statements create two comparative histograms, one for LENGTH and one for WIDTH:

```
proc capability data=steel;
   comphist / class = lot;
run;
```

Likewise, the following statements create comparative histograms for LENGTH and WIDTH:

```
proc capability data=steel;
   var length width;
   comphist / class = day;
run;
```

*In Release 6.12 and in previous releases of SAS/QC software, the keyword GRAPHICS was required in the PROC CAPABILITY statement since the COMPHISTOGRAM statement creates output only for high resolution graphics devices.

The following statements create three comparative histograms (for LENGTH, WIDTH, and LOT):

```
proc capability data=steel;
   comphist / class = day;
run;
```

The following statements create a comparative histogram for WIDTH only:

```
proc capability data=steel;
   var length width;
   comphist width / class=lot;
run;
```

class-variables
are one or two required classification variables. For example, the following statements create a one-way comparative histogram for WIDTH using the classification variable LOT:

```
proc capability data=steel;
   comphist width / class=lot;
run;
```

The following statements create a two-way comparative histogram for WIDTH classified by LOT and DAY:

```
proc capability data=steel;
   comphist width / class=(lot day);
run;
```

Note that the parentheses surrounding the *class-variables* are needed only if two classification variables are specified. See Output 10.1.1 on page 273 and Output 10.2.1 on page 275 for further examples.

options
control the features of the comparative histogram. All *options* are specified after the slash (/) in the COMPHIST statement. In the following example, the CLASS= option specifies the classification variable, the NORMAL option fits a normal density curve in each cell, and the CTEXT= option specifies the color of the text:

```
proc capability data=steel;
   comphist length / class = lot
                     normal
                     ctext = yellow;
run;
```

Summary of Options

The following tables list the COMPHIST statement options by function. For complete descriptions, see "Dictionary of Options" on page 257.

Normal Curve Options

Table 10.1 summarizes options that specify features of fitted normal distributions requested with the NORMAL option. Specify these options in parentheses after the NORMAL option.

Table 10.1. Normal-Options

COLOR=*color*	specifies color of normal curve
FILL	fills area under normal curve
L=*linetype*	specifies line type of normal curve
MU=*value*	specifies mean μ for fitted normal curve
SIGMA=*value*	specifies standard deviation σ for fitted normal curve
W=n	specifies width of normal curve

For example, the following statements use the NORMAL option to fit a normal curve in each cell of the comparative histogram:

```
proc capability;
   comphistogram / class = machine
                   normal(color=red l=2);
run;
```

The COLOR= *normal-option* draws the curve in red, and the L= *normal-option* specifies a line style of 2 (a dashed line) for the curve. In this example, maximum likelihood estimates are computed for the normal parameters μ and σ for each cell since these parameters are not specified.

Kernel Options

You can specify the options listed in Table 10.2 in parentheses after the keyword KERNEL to control features of kernel density estimates requested with the KERNEL option.

Table 10.2. Kernel Options

C=*value-list* | MISE	specifies standardized bandwidth parameter c for kernel density estimate
COLOR=*color*	specifies color of the kernel density curve
FILL	fills area under kernel density curve
K=*keyword*	specifies NORMAL, TRIANGULAR, or QUADRATIC kernel
L=*linetype*	specifies line type used for kernel density curve
LOWER=*value*	specifies lower bound for kernel density curve
UPPER=*value*	specifies upper bound for kernel density curve
W=n	specifies line width for kernel density curve

General Options

Table 10.3. Comparative Histogram Layout Options

ANNOKEY	applies annotation requested in ANNOTATE= data set to key cell only
BARLABEL=COUNT | PERCENT | PROPORTION	produces labels above histogram bars
BARWIDTH=*n*	specifies width for the bars
CLIPSPEC=CLIP | NOFILL	clips histogram bars at specification limits if there are no observations beyond the limits
ENDPOINTS=*values* | KEY | UNIFORM	labels interval endpoints and specifies how they are determined
HOFFSET=*value*	specifies offset for horizontal axis
INTERTILE=*value*	specifies distance between tiles
MAXNBIN=*n*	specifies maximum number of bins displayed
MAXSIGMAS=*value*	limits number of bins displayed to range of *value* standard deviations above and below mean of data in key cell
MIDPOINTS=*values* | KEY | UNIFORM	specifies how midpoints are determined
NCOLS=*n*	specifies number of columns in comparative histogram
NOBARS	suppresses histogram bars
NOFRAME	suppresses frame around plotting area
NOKEYMOVE	suppresses rearrangement of cells that occurs by default with the CLASSKEY= option
NOPLOT	suppresses plot
NROWS=*n*	specifies number of rows in comparative histogram
RTINCLUDE	includes right endpoint in interval
TILELEGLABEL='*string*'	specifies label displayed when _CTILE_ and _TILELG_ variables are provided in the CLASSSPEC= data set
TURNVLABELS	turns and strings out vertically characters in labels for vertical axis
WBARLINE=*n*	specifies line thickness for bar outlines

Table 10.4. Classification Options

CLASS=(*variables*)	specifies classification variables
CLASSKEY=(*'values'*)	specifies key cell
MISSING1	requests that missing values of first CLASS= variable be treated as a level of that CLASS= variable
MISSING2	requests that missing values of second CLASS= variable be treated as a level of that CLASS= variable
ORDER1=DATA | FORMATTED | FREQ | INTERNAL	specifies display order for values of the first CLASS= variable
ORDER2=DATA | FORMATTED | FREQ | INTERNAL	specifies display order for values of the second CLASS= variable

Table 10.5. Reference Line Options

FRONTREF	draws reference lines in front of histogram bars
HREF=*value-list*	specifies reference lines perpendicular to horizontal axis
HREFLABELS= *'label1'...'labeln'*	specifies labels for HREF= lines
HREFLABPOS=n	specifies vertical position of labels for HREF= lines
LHREF=*linetype*	specifies line style for HREF= lines
LVREF=*linetype*	specifies line style for VREF= lines
VREF=*value-list*	specifies reference lines perpendicular to vertical axis
VREFLABELS= *'label1'...'labeln'*	specifies labels for VREF= lines
VREFLABPOS=n	specifies horizontal position of labels for VREF= lines

Table 10.6. Text Enhancement Options

FONT=*font*	specifies software font for text
HEIGHT=*value*	specifies height of text used outside framed areas
INFONT=*font*	specifies software font for text inside framed areas
INHEIGHT=*value*	specifies height of text inside framed areas

Table 10.7. Axis and Legend Options

GRID	adds grid corresponding to vertical axis
LGRID=*linetype*	specifies line style for grid requested with GRID option
NLEGEND<='*string*'>	specifies form of the legend displayed inside tiles
NLEGENDPOS=NE | NW	specifies position of legend displayed inside tiles
NOHLABEL	suppresses label for horizontal axis
NOVLABEL	suppresses label for vertical axis
NOVTICK	suppresses tick marks and tick mark labels for vertical axis
VAXIS=*value-list*	specifies tick mark values for vertical axis
VAXISLABEL='*string*'	specifies label for vertical axis
VOFFSET=*value*	specifies length of offset at upper end of vertical axis
VSCALE=COUNT | PERCENT | PROPORTION	specifies scale for vertical axis
WAXIS=n	specifies line thickness for axes and frame
WGRID=n	specifies line thickness for grid

Table 10.8. Graphics Catalog Options

DESCRIPTION='*string*'	specifies description for graphics catalog member
NAME='*string*'	specifies name for plot in graphics catalog

Table 10.9. Color and Pattern Options

CAXIS=*color*	specifies color for axis
CBARLINE=*color*	specifies color for outline of the bars
CFILL=*color*	specifies color for filling bars
CFRAME=*color*	specifies color for frame
CFRAMENLEG=*color*	specifies the color for the frame requested by the NLEGEND option
CFRAMESIDE=*color*	specifies color for filling frame for row labels
CFRAMETOP=*color*	specifies color for filling frame for column labels
CGRID=*color*	specifies color for grid lines
CHREF=*color*	specifies color for HREF= lines
CPROP=*color*	specifies color for proportion of frequency bar
CTEXT=*color*	specifies color for text
CTEXTSIDE=*color*	specifies color for row labels
CTEXTTOP=*color*	specifies color for column labels
CVREF=*color*	specifies color for VREF= lines
PFILL=*pattern*	specifies pattern used to fill bars

Table 10.10. Input and Output Data Sets

ANNOTATE=*SAS-data-set*	annotate data set
CLASSSPEC=*SAS-data-set*	data set with specification limit information for each cell
OUTHISTOGRAM=*SAS-data-set*	information on histogram intervals

Dictionary of Options

The following entries describe the *options* in detail. All options apply with high resolution graphics output.

ANNOKEY

specifies that annotation requested with the ANNOTATE= option is to be applied only to the *key cell*. By default, annotation is applied to all of the cells. Use the CLASSKEY= option to specify the key cell.

ANNOTATE=*SAS-data-set*
ANNO=*SAS-data-set*

specifies an input data set containing annotate variables as described in *SAS/GRAPH Software: Reference*. You can use this data set to add features to the comparative histogram. The ANNOTATE= data set you specify in the COMPHISTOGRAM statement is used for all plots created by the statement. You can also specify an ANNOTATE= data set in the PROC CAPABILITY statement to enhance all plots created by the procedure; for more information, see "ANNOTATE= Data Sets" on page 189.

BARLABEL=COUNT | PERCENT | PROPORTION

displays labels above the histogram bars. If you specify BARLABEL=COUNT, the label shows the number of observations associated with a given bar. BARLABEL=PERCENT shows the percent of observations represented by that bar. If you specify BARLABEL=PROPORTION, the label displays the proportion of observations associated with the bar.

BARWIDTH=*value*

specifies the width of the histogram bars in screen percent units.

C=*value-list* **| MISE**

specifies the standardized bandwidth parameter c for kernel density estimates requested with the KERNEL option. You can specify up to five *values* to display multiple estimates in each cell. You can also specify the keyword MISE to request the bandwidth parameter that minimizes the estimated mean integrated square error (MISE). For example, consider the following statements (for more information, see "Kernel Density Estimates" on page 319):

```
proc capability;
   comphist length / class=batch kernel(c = 0.5 1.0 mise);
run;
```

The KERNEL option displays three density estimates. The first two have standardized bandwidths of 0.5 and 1.0, respectively. The third has a bandwidth parameter that

minimizes the MISE. You can also use the C= and K= options (K= specifies kernel type) to display multiple estimates. For example, consider the following statements:

```
proc capability;
   comphist length / class = batch
                     kernel(c = 0.75 k = normal triangular);
run;
```

Here two estimates are displayed. The first uses a normal kernel and bandwidth parameter of 0.75, and the second uses a triangular kernel and a bandwidth parameter of 0.75. In general, if more kernel types are specified than bandwidth parameters, the last bandwidth parameter in the list will be repeated for the remaining estimates. Likewise, if more bandwidth parameters are specified than kernel types, the last kernel type will be repeated for the remaining estimates. The default is MISE.

CAXIS=*color*
CAXES=*color*
CA=*color*
specifies the color for the axes, tick marks, and target line. The default is the first color in the device color list. This color is also used for grid lines, unless overridden by the CGRID= option.

CBARLINE=*color*
specifies the color of the outline of the histogram bars. This option overrides the C= option in the SYMBOL1 statement. The default is the first color in the device color list.

CFILL=*color*
specifies a color used to fill the bars of the histograms (or the areas under a fitted curve if you also specify the FILL option). See the entry for the FILL option for additional details. See Output 10.1.1 on page 273 and Example 10.2 on page 274 for examples. Refer to *SAS/GRAPH Software: Reference* for a list of colors. By default, bars and curve areas are not filled.

CFRAME=*color*
specifies the color for the area enclosed by the axes and the frame. This area is not filled by default. The CFRAME= option cannot be used with the NOFRAME option, the CTILES= option, or the variable _CTILE_ in a CLASSSPEC= data set.

CFRAMENLEG=*color* **| EMPTY**
specifies that the legend requested with the NLEGEND option (or the variable _TILELB_ in a CLASSSPEC= data set) is to be framed and that the frame is to be filled with the color indicated. If you specify CFRAMENLEG=EMPTY, a frame is drawn but not filled with a color.

CFRAMESIDE=*color*
specifies the color for filling the frame area for the row labels displayed along the left side of a comparative histogram requested with the CLASS= option. This color is also used to fill the frame area for the label of the corresponding CLASS= variable (if a label is associated with the variable.) See Output 10.2.1 on page 275 for an example. By default, these areas are not filled.

CFRAMETOP=*color*
specifies the color for filling the frame area for the column labels displayed across the top of a comparative histogram requested with the CLASS= option. This color is also used to fill the frame area for the label of the corresponding CLASS= variable (if a label is associated with the variable.) See Output 10.2.1 on page 275 for an example. By default, these areas are not filled.

CGRID=*color*
specifies the color for grid lines requested with the GRID option. By default, grid lines are the same color as the axes. If you use CGRID=, you do not need to specify the GRID option.

CHREF=*color*
specifies the color for lines requested with the HREF= option. The default is the first color in the device color list.

CLASS=*variable*
CLASS=(*variable1 variable2***)**
specifies that a comparative histogram is to be created using the levels of the *variables* (also referred to as *class-variables* or *classification variables*).

If you specify a single *variable*, a one-way comparative histogram is created. The observations in the input data set are sorted by the formatted values (levels) of the variable. A separate histogram is created for the process variable values in each level, and these component histograms are arranged in an array to form the comparative histogram. Uniform horizontal and vertical axes are used to facilitate comparisons. For an example, see Figure 10.1 on page 249.

If you specify two *classification variables*, a two-way comparative histogram is created. The observations in the input data set are cross-classified according to the values (levels) of these variables. A separate histogram is created for the process variable values in each cell of the cross-classification, and these component histograms are arranged in a matrix to form the comparative histogram. The levels of *variable1* are used to label the rows of the matrix, and the levels of *variable2* are used to label the columns of the matrix. Uniform horizontal and vertical axes are used to facilitate comparisons. For an example, see Output 10.2.1 on page 275.

Classification variables can be numeric or character. Formatted values are used to determine the levels. You can specify whether missing values are to be treated as a level with the MISSING1 and MISSING2 options.

If a label is associated with a classification variable, the label is displayed on the comparative histogram. The variable label is displayed parallel to the column (or row) labels. For an example, see Figure 10.1 on page 249.

CLASSKEY=*'value'*
CLASSKEY=(*'value1' 'value2'***)**
specifies the *key cell* in a comparative histogram requested with the CLASS= option. The bin size and midpoints are first determined for the key cell, and then the midpoint list is extended to accommodate the data ranges for the remaining cells. Thus, the choice of the key cell determines the uniform horizontal axis used for all cells.

If you specify CLASS=*variable*, you can specify CLASSKEY=*'value'* to identify the key cell as the level for which *variable* is equal to *value*. You must specify a formatted *value*. By default, the levels are sorted in the order determined by the ORDER1= option, and the key cell is the level that occurs first in this order. The cells are displayed in this order from top to bottom (or left to right), and, consequently, the key cell is displayed at the top or at the left. If you specify a different key cell with the CLASSKEY= option, this cell is displayed at the top or at the left unless you also specify the NOKEYMOVE option.

If you specify CLASS=(*variable1 variable2*), you can specify CLASSKEY=(*'value1' 'value2'*) to identify the key cell as the level for which *variable1* is equal to *value1* and *variable2* is equal to *value2*. Here, *value1* and *value2* must be formatted values, and they must be enclosed in quotes. For an example of the CLASSKEY= option with a two-way comparative histogram, see Output 10.2.1 on page 275. By default, the levels of *variable1* are sorted in the order determined by the ORDER1= option, and within each of these levels, the levels of *variable2* are sorted in the order determined by the ORDER2= option. The default key cell is the combination of levels of *variable1* and *variable2* that occurs first in this order. The cells are displayed in order of *variable1* from top to bottom and in order of *variable2* from left to right. Consequently, the default key cell is displayed in the upper left corner. If you specify a different key cell with the CLASSKEY= option, this cell is displayed in the upper left corner unless you also specify the NOKEYMOVE option.

CLASSSPEC=*SAS-data-set*
CLASSSPECS=*SAS-data-set*

specifies a data set that provides distinct specification limits for each cell, as well as a color, legend, and label for the corresponding tile. The following table lists the variables that are read from a CLASSSPECS= data set:

Variable Name	Description
BY variables	subsets the data set
Classification variables	specifies the structure of the comparative histogram
VAR	specifies name of process variable (must be character variable of length 8)
LSL	specifies lower specification limit for tile
TARGET	specifies target value for tile
USL	specifies upper specification limit for tile
CTILE	specifies background color for tiles (must be character variable of length 8)
TILELG	specifies text displayed in color tile legend at bottom of comparative histogram (character variable of length not greater than 16)
TILELB	specifies text displayed in corner of each tile (character variable of length not greater than 16)

If you specify a CLASSSPEC= data set, you cannot use the SPEC statement or a SPEC= data set. If you use a BY statement, the CLASSSPEC= data set must contain one observation for each unique combination of process and classification variables within each BY group. See Example 10.1 on page 272 for an example of a CLASSSPECS= data set.

Also note that

- you can suppress the background color for a tile by assigning the value `EMPTY` or a blank value to the variable _CTILE_
- you can use the NLEGENDPOS= option to specify the corner of the tile in which the _TILELB_ label is displayed. You can frame the label with the CFRAMENLEG= option.
- you cannot use the variable _TILELG_ unless you specify the variable _CTILE_
- the variable _TILELB_ takes precedence over the NLEGEND option

CLIPSPEC=CLIP | NOFILL
specifies that histogram bars are clipped at the upper and lower specification limit lines when there are no observations outside the specification limits. The bar intersecting the lower specification limit is clipped if there are no observations less than the lower limit; the bar intersecting the upper specification limit is clipped if there are no observations greater than the upper limit. If you specify CLIPSPEC=CLIP, the histogram bar is truncated at the specification limit. If you specify CLIPSPEC=NOFILL, the portion of a filled histogram bar outside the specification limit is left unfilled. Specifying CLIPSPEC=NOFILL when histogram bars are not filled has no effect.

COLOR=*color*
specifies the color of the normal density curve or the kernel density estimate curve. Enclose the COLOR= option in parentheses after the NORMAL option or the KERNEL option. See Output 10.1.1 on page 273 for an example.

CPROP=*color*
specifies the color for a horizontal bar whose length (relative to the width of the tile) indicates the proportion of the total frequency that is represented by the corresponding cell. For an example, see Figure 10.2 on page 250. Empty bars are displayed if you specify CPROP=EMPTY. By default, bars are not displayed.

CTEXT=*color*
CT=*color*
specifies the color for tick mark labels and axis labels. The default is the color specified for the CTEXT= option in the most recent GOPTIONS statement.

CTEXTSIDE=*color*
specifies the color of the row labels displayed along the left side of a comparative histogram. By default, the CTEXT= color is used for these labels.

CTEXTTOP=*color*
specifies the color of the column labels displayed along the top of a comparative histogram. By default, the CTEXT= color is used for these labels.

CVREF=*color*
specifies the color for lines requested with the VREF= option. The default is the first color in the device color list.

DESCRIPTION=*'string'*
DES=*'string'*
specifies a description, up to 40 characters, that appears in the PROC GREPLAY master menu. The default is the variable name.

ENDPOINTS=*value-list* **| KEY | UNIFORM**
specifies that histogram interval endpoints, rather than midpoints, are aligned with horizontal axis tick marks, and specifies how the endpoints are determined. The method you specify is used for all process variables analyzed with the COMPHISTOGRAM statement.

If you specify ENDPOINTS=*value-list*, the *values* must be listed in increasing order and must be evenly spaced. The difference between consecutive endpoints is used as the width of the histogram bars. The first value is the lower bound of the first histogram bin and the last value is the upper bound of the last bin. Thus, the number of values in the list is one greater than the number of bins it specifies. If the range of the *values* does not cover the range of the data as well as any specification limits (LSL and USL) that are given, the list is extended in either direction as necessary.

If you specify ENDPOINTS=KEY, the procedure first determines the endpoints for the data in the key cell. The initial number of endpoints is based on the number of observations in the key cell using the method of Terrell and Scott (1985). The endpoint list for the key cell is then extended in either direction as necessary until it spans the data in the remaining cells. If the key cell contains no observations, the method of determining bins reverts to ENDPOINTS=UNIFORM.

If you specify ENDPOINTS=UNIFORM, the procedure determines the endpoints using all the observations as if there were no cells. In other words, the number of endpoints is based on the total sample size using the method of Terrell and Scott (1985).

FILL
fills areas under a fitted density curve with colors and patterns. Enclose the FILL option in parentheses after the keyword NORMAL or KERNEL. Depending on the area to be filled (outside or between the specification limits), you can specify the color and pattern with options in the SPEC statement and the COMPHISTOGRAM statement, as summarized in the following table:

Area Under Curve	Statement	Option
between specification limits	COMPHIST COMPHIST	CFILL=*color* PFILL=*pattern*
left of lower specification limit	SPEC SPEC	CLEFT=*color* PLEFT=*pattern*
right of upper specification limit	SPEC SPEC	CRIGHT=*color* PRIGHT=*pattern*

If you do not display specification limits, you can use the CFILL= and PFILL= options to specify the color and pattern for the entire area under the curve. Solid fills are used by default if patterns are not specified. You can specify the FILL option with only one fitted curve. For an example, see Output 10.1.1 on page 273. Refer to *SAS/GRAPH Software: Reference* for a list of available patterns and colors. If you do not specify the FILL option but you do specify the options in the preceding table, the colors and patterns are applied to the corresponding areas under the histogram.

FONT=*font*
specifies a software font for text used outside the framed areas of a comparative histogram (labels for axes, tick marks, and so forth). This font takes precedence over the FTEXT= font specified in a GOPTIONS statement. Refer to *SAS/GRAPH Software: Reference* for a list of fonts.

FRONTREF
draws reference lines requested with the HREF= and VREF= options in front of the histogram bars. By default, reference lines are drawn behind the histogram bars and can be obscured by them. *Graphics*

GRID
adds a grid to the comparative histogram. Grid lines are horizontal lines positioned at major tick marks on the vertical axis.

HEIGHT=*value*
specifies the height in percent screen units of text for axis labels, tick mark labels, and legends. The HEIGHT= option takes precedence over the HTEXT= option in the GOPTIONS statement.

HOFFSET=*value*
specifies the offset in percent screen units at both ends of the horizontal axis. Specify HOFFSET=0 to eliminate the default offset.

HREF=*value-list*
draws reference lines perpendicular to the horizontal axis at the values specified. For an illustration, see Output 11.1.1 on page 335.

HREFLABELS=*'label1'...'labeln'*
HREFLABEL=*'label1'...'labeln'*
HREFLAB=*'label1'...'labeln'*
specifies labels for the lines requested with the HREF= option. The number of labels

must equal the number of lines. Enclose the labels in quotes. Labels can be up to 16 characters. For an illustration, see Output 11.1.1 on page 335.

HREFLABPOS=*n*
specifies the vertical position of HREFLABELS= labels as follows: 1 positions the labels along the top of the histogram; 2 staggers the labels from top to bottom; 3 positions the labels along the bottom. The default is 1.

INFONT=*font*
specifies a software font for text used inside the framed areas of the comparative histogram (such as sample size legends). The INFONT= option takes precedence over the FTEXT= option in the GOPTIONS statement. Refer to *SAS/GRAPH Software: Reference* for a list of fonts.

INHEIGHT=*value*
specifies the height in percent screen units of text used inside the framed areas of the comparative histogram (such as sample size legends). The default height is the height you specify with the HEIGHT= option. If you do not specify the HEIGHT= option, the default height is the height you specify with the HTEXT= option in the GOPTIONS statement.

INTERTILE=*value*
specifies the distance in horizontal percent screen units between tiles. For an example, see Figure 10.2 on page 250. By default, the tiles are contiguous.

K=NORMAL | TRIANGULAR | QUADRATIC
specifies the type of kernel (normal, triangular, or quadratic) used to compute kernel density estimates requested with the KERNEL option. Enclose the K= option in parentheses after the keyword KERNEL. You can specify a single type or a list of types. If you specify more estimates than types, the last kernel type in the list is used for the remaining estimates. By default, a normal kernel is used.

KERNEL<**(** *kernel-options* **)**>
requests a kernel density estimate for each cell of the comparative histogram. You can specify the *kernel-options* described in the following table:

FILL	specifies that the area under the curve is to be filled
COLOR=	specifies the color of the curve
L=	specifies the line style for the curve
W=	specifies the width of the curve
K=	specifies the type of kernel
C=	specifies the smoothing parameter
LOWER=	specifies the lower bound for the curve
UPPER=	specifies the upper bound for the curve

See Output 10.1.1 on page 273 for an example. By default, the estimate is based on the AMISE method. For more information, see "Kernel Density Estimates" on page 319.

L=*linetype*
specifies the line type for a normal or kernel density estimate curve. Enclose the L= option in parentheses after the NORMAL option or the KERNEL option. If you use the L= option with the KERNEL option, you can specify a single line type or a list of line types. Refer to *SAS/GRAPH Software: Reference* for a list of available line types. The default is 1, which produces a solid line.

LOWER=*value*
specifies the lower bound for a kernel density estimate curve. Enclose the LOWER= option in parentheses after the KERNEL option. You can specify a single lower bound or a list of lower bounds. By default, a kernel density estimate curve has no lower bound.

LGRID=*n*
specifies the line type for the grid requested with the GRID option. If you use the LGRID= option, you do not need to specify the GRID option. The default is 1, which produces a solid line.

LHREF=*n*
LH=*n*
specifies the line type for lines requested with the HREF= option. The default is 2, which produces a dashed line.

LVREF=*n*
LV=*n*
specifies the line type for lines requested with the VREF= option. The default is 2, which produces a dashed line.

MAXNBIN=*n*
specifies the maximum number of bins to be displayed. This option is useful in situations where the scales or ranges of the data distributions differ greatly from cell to cell. By default, the bin size and midpoints are determined for the key cell, and then the midpoint list is extended to accommodate the data ranges for the remaining cells. However, if the cell scales differ considerably, the resulting number of bins may be so great that each cell histogram is scaled into a narrow region. By limiting the number of bins with the MAXNBIN= option, you can narrow the window about the data distribution in the key cell. Note that the MAXNBIN= option provides an alternative to the MAXSIGMAS= option.

MAXSIGMAS=*value*
limits the number of bins to be displayed to a range of *value* standard deviations (of the data in the key cell) above and below the mean of the data in the key cell. This option is useful in situations where the scales or ranges of the data distributions differ greatly from cell to cell. By default, the bin size and midpoints are determined for the key cell, and then the midpoint list is extended to accommodate the data ranges for the remaining cells. If the cell scales differ considerably, however, the resulting number of bins may be so great that each cell histogram is scaled into a narrow region. By limiting the number of bins with the MAXSIGMAS= option, you narrow the window about the data distribution in the key cell. Note that the MAXSIGMAS= option provides an alternative to the MAXNBIN= option.

MIDPOINTS=*value-list* **| KEY | UNIFORM**
specifies how midpoints are determined for the bins in the comparative histogram. The method you specify is used for all process variables analyzed with the COMPHISTOGRAM statement.

If you specify MIDPOINTS=*value-list*, the *values* must be listed in increasing order and must be evenly spaced. The difference between consecutive midpoints is used as the width of the histogram bars. If the range of the *values* does not cover the range of the data as well as any specification limits (LSL and USL) that are given, the list is extended in either direction as necessary. See Example 10.1 on page 272 for an illustration.

If you specify MIDPOINTS=KEY, the procedure first determines the midpoints for the data in the key cell. The initial number of midpoints is based on the number of observations in the key cell using the method of Terrell and Scott (1985). The midpoint list for the key cell is then extended in either direction as necessary until it spans the data in the remaining cells.

If you specify MIDPOINTS=UNIFORM, the procedure determines the midpoints using all the observations as if there were no cells. In other words, the number of midpoints is based on the total sample size using the method of Terrell and Scott (1985).

By default, MIDPOINTS=KEY. However, if the key cell contains no observations, the default is MIDPOINTS=UNIFORM.

MISSING1
specifies that missing values of the first CLASS= variable are to be treated as a level of the CLASS= variable. If the first CLASS= variable is a character variable, a missing value is defined as a blank internal (unformatted) value. If the process variable is numeric, a missing value is defined as any of the SAS System missing values. If you do not specify MISSING1, observations for which the first CLASS= variable is missing are excluded from the analysis.

MISSING2
specifies that missing values of the second CLASS= variable are to be treated as a level of the CLASS= variable. If the second CLASS= variable is a character variable, a missing value is defined as a blank internal (unformatted) value. If the process variable is numeric, a missing value is defined as any of the SAS System missing values. If you do not specify MISSING2, observations for which the second CLASS= variable is missing are excluded from the analysis.

MU=*value*
specifies the parameter μ for the normal density curves requested with the NORMAL option. Enclose the MU= option in parentheses after the NORMAL option. The default value is the sample mean of the observations in the cell.

NAME=*'string'*
specifies a name for the plot, up to eight characters, that appears in the PROC GREPLAY master menu. The default is 'CAPABILI'.

NCOLS=n
NCOL=n
specifies the number of columns in a comparative histogram. You can use the NCOLS= option with the NROWS= option if you specify two CLASS= variables. See Output 10.2.1 on page 275 for an example of a two-way comparative histogram using the NCOLS= option. By default, NCOLS=1 (and NROWS=2) if you specify only one CLASS= variable, and NCOLS=2 (and NROWS=2) if you specify two CLASS= variables.

NLEGEND<**=***'label'*>
specifies the form of a legend that is displayed inside each tile and indicates the sample size of the cell. The following two forms are available:

- If you specify the NLEGEND option, the form is $N = n$ where n is the cell sample size.
- If you specify the NLEGEND='*label*' option, the form is *label* $= n$ where n is the cell sample size. The label can be up to 16 characters and must be enclosed in quotes. For instance, you might specify NLEGEND='Number of Parts' to request a label of the form *Number of Parts* $= n$.

See Figure 10.1 on page 249 for an example. You can use the CFRAMENLEG= option to frame the sample size legend. The variable _TILELB_ in a CLASSSPECS= data set overrides the NLEGEND option. By default, no legend is displayed.

NLEGENDPOS=NW | NE
specifies the position of the legend requested with the NLEGEND option or the variable _TILELB_ in a CLASSSPEC= data set. If NLEGENDPOS=NW, the legend is displayed in the northwest corner of the tile; if NLEGENDPOS=NE, the legend is displayed in the northeast corner of the tile. See Figure 10.1 on page 249 for an illustration. The default is NE.

NOBARS
suppresses the display of the bars in a comparative histogram.

NOCHART
suppresses the creation of a comparative histogram. This is an alias for NOPLOT.

NOFRAME
suppresses the frame around each tile. The NOFRAME option cannot be specified with the CFRAME= option.

NOHLABEL
suppresses the label for the horizontal axis. This is useful for avoiding clutter.

NOKEYMOVE
suppresses the rearrangement of cells that occurs by default when you use the CLASSKEY= option to specify the key cell. For details, see the entry for the CLASSKEY= option.

NOPLOT
suppresses the creation of a comparative histogram. This option is useful when you are using the COMPHISTOGRAM statement solely to create an output data set.

NORMAL<**(***normal-options***)**>
displays a normal density curve for each cell of the comparative histogram. The equation of the normal density curve is

$$p(x) = \frac{h \times 100\%}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \quad \text{for } -\infty < x < \infty$$

where

μ = mean
σ = standard deviation ($\sigma > 0$)
h = width of histogram interval

If you specify values for μ and σ with the MU= and SIGMA= *normal-options*, the same curve is displayed for each cell. By default, a distinct curve is displayed for each cell based on the sample mean and standard deviation for that cell. For example, the following statements display a distinct curve for each level of the variable SUPPLIER:

```
proc capability noprint;
   comphist width / class=supplier normal(color=red l=2);
run;
```

The curves are drawn in red with a line style of 2 (a dashed line). See Figure 10.2 on page 250 for another illustration. Table 10.1 on page 253 lists options that can be specified in parentheses after the NORMAL option.

NOVLABEL
suppresses the label for the vertical axis.

NOVTICK
suppresses the tick marks and tick mark labels for the vertical axis. If you specify the NOVTICK option, the NOVLABEL option is assumed.

NROWS=n
NROW=n
specifies the number of rows in a comparative histogram. You can use the NROWS= option with the NCOLS= option if you specify two CLASS= variables. See Figure 10.1 on page 249 for a *one-way* comparative histogram using the NROWS= option, and see Output 10.2.1 on page 275 for a *two-way* comparative histogram using the NROWS= and NCOLS= options. The default is 2.

ORDER1=INTERNAL | FORMATTED | DATA | FREQ
specifies the display order for the values of the first CLASS= variable.

The levels of the first CLASS= variable are always constructed using the *formatted* values of the variable, and the formatted values are always used to label the rows

(columns) of a comparative histogram. You can use the ORDER1= option to determine the order of the rows (columns) corresponding to these values, as follows:

- **If you specify ORDER1=INTERNAL**, the rows (columns) are displayed from top to bottom (left to right) in increasing order of the internal (unformatted) values of the first CLASS= variable. If there are two or more distinct internal values with the same formatted value, then the order is determined by the internal value that occurs first in the input data set.

  For example, suppose that you specify a numeric CLASS= variable called DAY (with values 1, 2, and 3). Suppose also that a format (created with the FORMAT procedure) is associated with DAY and that the formatted values are as follows: 1 = 'Wednesday', 2 = 'Thursday', and 3 = 'Friday'. If you specify ORDER1=INTERNAL, the rows of the comparative histogram will appear in day-of-the-week order (*Wednesday*, *Thursday*, *Friday*) from top to bottom.
- **If you specify ORDER1=FORMATTED**, the rows (columns) are displayed from top to bottom (left to right) in increasing order of the formatted values of the first CLASS= variable. In the preceding illustration, if you specify ORDER1=FORMATTED, the rows will appear in alphabetical order (*Friday*, *Thursday*, *Wednesday*) from top to bottom.
- **If you specify ORDER1=DATA**, the rows (columns) are displayed from top to bottom (left to right) in the order in which the values of the first CLASS= variable first appear in the input data set.
- **If you specify ORDER1=FREQ**, the rows (columns) are displayed from top to bottom (left to right) in order of *decreasing* frequency count. If two or more classes have the same frequency count, the order is determined by the formatted values.

By default, ORDER1=INTERNAL.

ORDER2=INTERNAL | FORMATTED | DATA | FREQ

specifies the display order for the values of the second CLASS= variable.

The levels of the second CLASS= variable are always constructed using the *formatted* values of the variable, and the formatted values are always used to label the columns of a two-way comparative histogram. You can use the ORDER2= option to determine the order of the columns.

The layout of a two-way comparative histogram is determined by using the ORDER1= option to obtain the order of the rows from top to bottom (recall that ORDER1=INTERNAL by default). Then the ORDER2= option is applied to the observations corresponding to the first row to obtain the order of the columns from left to right. If any columns remain unordered (that is, the categories are *unbalanced*), the ORDER2= option is applied to the observations in the second row, and so on, until all the columns have been ordered.

The values of the ORDER2= option are interpreted as described for the ORDER1= option. By default, ORDER2=INTERNAL.

OUTHISTOGRAM=*SAS-data-set*
creates a SAS data set that saves the midpoints or endpoints of the histogram intervals, the observed percent of observations in each interval, and (optionally) the percent of observations in each interval estimated from a fitted normal distribution. By default, interval midpoint values are saved in the variable _MIDPT_. If the ENDPOINTS= option is specified, intervals are identified by endpoint values instead. If RTINCLUDE is specified, the _MAXPT_ variable contains upper endpoint values. Otherwise, lower endpoint values are saved in the _MINPT_ variable.

PFILL=*pattern*
specifies a pattern used to fill the bars of the histograms (or the areas under a fitted curve if you also specify the FILL option). See the entries for the CFILL= and FILL options for additional details. Refer to *SAS/GRAPH Software: Reference* for a list of pattern values. By default, the bars and curve areas are not filled.

RTINCLUDE
includes the right endpoint of each histogram interval in that interval. The left endpoint is included by default.

SIGMA=*value*
specifies the parameter σ for normal density curves requested with the NORMAL option. Enclose the SIGMA= option in parentheses after the NORMAL option. The default value is the sample standard deviation of the observations in the cell.

TILELEGLABEL='*label*' specifies a label displayed to the left of the legend that is created when you provide _CTILE_ and _TILELG_ variables in a CLASSSPEC= data set. The *label* can be up to 16 characters and must be enclosed in quotes. The default *label* is *Tiles:*.

TURNVLABEL
TURNVLABELS
specifies that the characters in the labels for the vertical axis are to be turned and strung out vertically. This happens by default when a hardware font is used.

UPPER=*value*
specifies the upper bound for a kernel density estimate curve. Enclose the UPPER= option in parentheses after the KERNEL option. You can specify a single upper bound or a list of upper bounds. By default, a kernel density estimate curve has no upper bound.

VAXIS=*value-list*
specifies tick mark values for the vertical axis. The values must be equally spaced and in increasing order, and the first value must be zero. You must scale the values in the same units as the bars (see the VSCALE= option), and the last value must be greater than or equal to the height of the largest bar. See Output 10.2.1 on page 275 for an example.

VAXISLABEL=*'label'*
specifies a label (up to 40 characters) for the vertical axis.

VOFFSET=*value*
specifies the offset in percent screen units at the upper end of the vertical axis.

VREF=*value-list*
draws reference lines perpendicular to the vertical axis at the values specified. For an illustration, see Output 9.2.1 on page 244.

VREFLABELS=*'label1'...'labeln'*
VREFLABEL=*'label1'...'labeln'*
VREFLAB=*'label1'...'labeln'*
specifies labels for the lines requested with the VREF= option. The number of labels must equal the number of lines. Enclose the labels in quotes. Labels can be up to 16 characters. For an illustration, see Output 9.2.1 on page 244.

VREFLABPOS=*n*
specifies the horizontal position of VREFLABELS= labels as follows: VREFLABPOS=1 positions the labels at the left of the tile, and VREFLABPOS=2 positions the labels at the right. The default is 1.

VSCALE=PERCENT | COUNT | PROPORTION
specifies the scale of the vertical axis. The value COUNT scales the data in units of the number of observations per data unit. The value PERCENT scales the data in units of percent of observations per data unit. The value PROPORTION scales the data in units of proportion of observations per data unit. The default is PERCENT.

W=*n*
specifies the width in pixels of the curve. Enclose the W= option in parentheses after the NORMAL option or the KERNEL option. The default is 1.

WAXIS=*n*
specifies the line thickness (in pixels) for the axes and frame. The default is 1. This thickness is also used for grid lines, unless overridden by the WGRID= option.

WBARLINE=*n*
specifies the width of bar outlines. By default, $n = 1$.

WGRID=*n*
specifies the width of the grid lines requested with the GRID option. By default, grid lines are the same width as the axes. If you use the WGRID= option, you do not need to specify the GRID option.

Examples

This section provides advanced examples of comparative histograms.

Example 10.1. Adding Insets with Descriptive Statistics

See CAPCMH2 in the SAS/QC Sample Library

Three similar machines are used to attach a part to an assembly. One hundred assemblies are sampled from the output of each machine, and a part position is measured in millimeters. The following statements save the measurements in a SAS data set named MACHINES:

```
data machines;
   input position @@;
   label position='Position in Millimeters';
   if      (_n_ <= 100) then machine = 'Machine 1';
   else if (_n_ <= 200) then machine = 'Machine 2';
   else                      machine = 'Machine 3';
   datalines;
-0.17  -0.19  -0.24  -0.24  -0.12   0.07  -0.61   0.22   1.91  -0.08
-0.59   0.05  -0.38   0.82  -0.14   0.32   0.12  -0.02   0.26   0.19
-0.07   0.13  -0.49   0.07   0.65   0.94  -0.51  -0.61  -0.57  -0.51
 0.01  -0.51   0.07  -0.16  -0.32  -0.42  -0.42  -0.34  -0.34  -0.35
-0.49   0.11  -0.42   0.76   0.02  -0.59  -0.28   1.12  -0.02  -0.60
-0.64   0.13  -0.32  -0.77  -0.02  -0.07  -0.49  -0.53  -0.22   0.61

  ...

 0.09   0.78   0.46  -0.13   0.69   0.66   0.29   0.25   0.54   1.03
 0.53   0.72   0.99   0.84   0.35   0.67   0.91   0.36   1.06   0.44
 0.58   0.46   0.58   0.92   0.70   0.81   0.07   0.33   0.82   0.62
 0.48   0.41   0.78   0.58   0.43   0.07   0.27   0.49   0.79   0.92
 0.79   0.66   0.22   0.71   0.53   0.57   0.90   0.48   1.17   1.03
;
run;
```

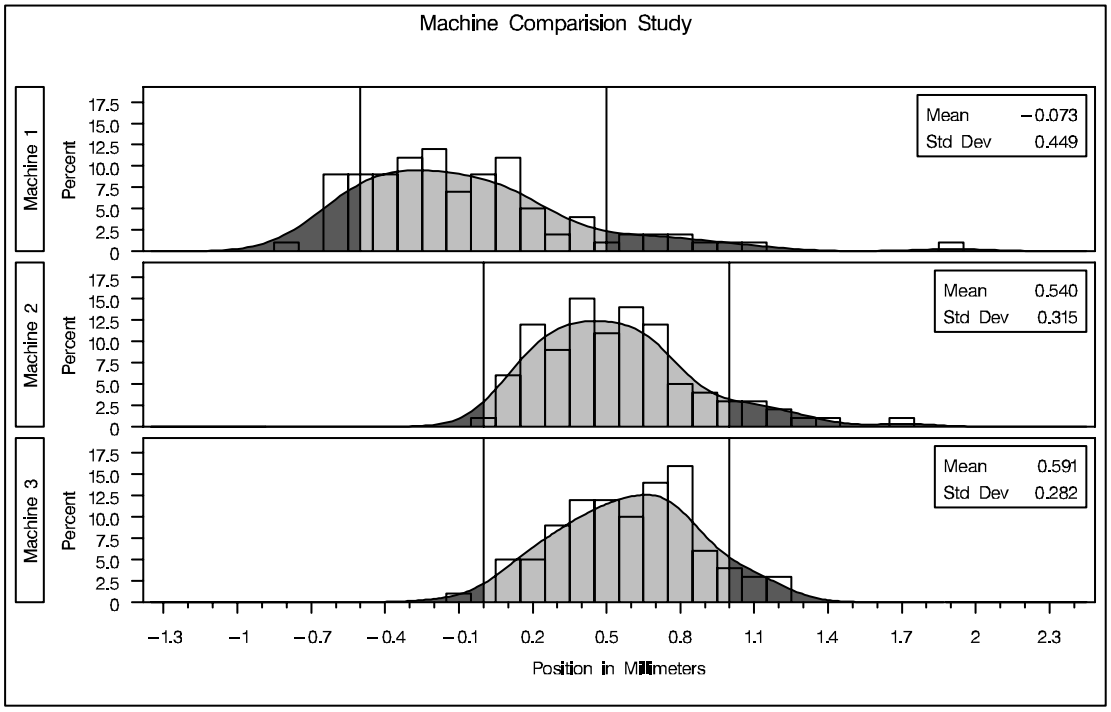
Distinct specification limits for the three machines are provided in a data set named SPECLIMS.

```
data speclims;
   input machine $9. _lsl_ _usl_;
   _var_ = 'position';
   datalines;
Machine 1 -0.5 0.5
Machine 2  0.0 1.0
Machine 3  0.0 1.0
;
run;
```

The following statements create a comparative histogram for the measurements in MACHINES that displays the specification limits in SPECLIMS. The display is shown in Output 10.1.1.

```
title 'Machine Comparision Study';
proc capability data=machines noprint;
   spec clsl=black cusl=black cleft=dagr cright=dagr;
   comphist position / class      = machine
                       nrows      = 3
                       intertile  = 1
                       midpoints  = -1.2 to 2.2 by 0.1
                       kernel (color=black fill)
                       cfill      = ligr
                       classspecs = speclims;
   inset mean std="Std Dev" / pos = ne format = 6.3;
run;
```

Output 10.1.1. Comparative Histograms



The INSET statement is used to inset the sample mean and standard deviation for each machine in the corresponding tile. The MIDPOINTS= option specifies the midpoints of the histogram bins. Kernel density estimates are displayed using the KERNEL option. The curve areas outside the specification limits are filled using the CLEFT= and CRIGHT= options in the SPEC statement, and the area between the limits is filled using the CFILL= option in COMPHISTOGRAM statement.

Example 10.2. Creating a Two-Way Comparative Histogram

See CAPCMH3 in the SAS/QC Sample Library

Two suppliers (A and B) provide disk drives for a computer manufacturer. The manufacturer measures the disk drive opening width to compare the process capabilities of the suppliers and determine whether there has been an improvement from 1992 to 1993.

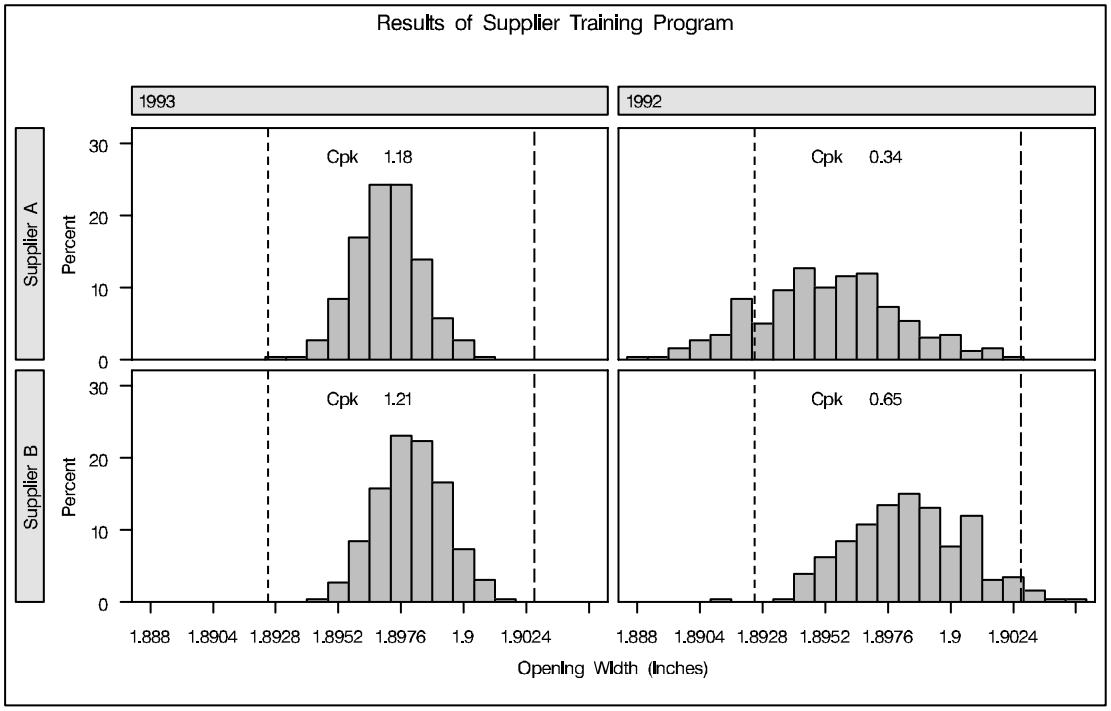
The following statements save the measurements in a data set named DISK. There are two classification variables, SUPPLIER and YEAR, and a format is associated with YEAR.

```
proc format ;
   value mytime  1 = '1992'
                 2 = '1993' ;

data disk;
   input @1 supplier $10. year width;
   label width = 'Opening Width (inches)';
   format year mytime.;
   datalines;
Supplier A   1   1.8932
Supplier A   1   1.8952
    .        .    .
    .        .    .
Supplier B   1   1.8980
Supplier B   1   1.8986
Supplier A   2   1.8978
Supplier A   2   1.8966
    .        .    .
    .        .    .
Supplier B   2   1.8967
Supplier B   2   1.8997
;
```

The following statements create the comparative histogram in Output 10.2.1:

```
title "Results of Supplier Training Program";
proc capability data=disk noprint;
   specs  lsl = 1.8925  llsl = 2
          usl = 1.9027  lusl = 3 ;
   comphist width / class      = ( supplier year )
                    classkey   = ('Supplier A' '1993')
                    intertile  = 1.0
                    vaxis      = 0 10 20 30
                    ncols      = 2
                    nrows      = 2
                    cfill      = ligr
                    cframetop  = yellow
                    cframeside = yellow;
   inset cpk (4.2) / noframe pos = n;
run;
```

Output 10.2.1. Two-Way Comparative Histogram



The CLASSKEY= option specifies the key cell as the observations for which SUPPLIER is equal to `SUPPLIER A` and YEAR is equal to `2`. This cell determines the binning for the other cells, and (since the NOKEYMOVE option is not specified) the columns are interchanged so that this cell is displayed in the upper left corner. Note that if the CLASSKEY= option were not specified, the default key cell would be the observations for which SUPPLIER is equal to `SUPPLIER A` and YEAR is equal to `1`. If the CLASSKEY= option were not specified (or if the NOKEYMOVE option were specified), the column labeled *1992* would be displayed to the left of the column labeled *1993*. See the entry for the CLASSKEY= option on page 259 for details.

The VAXIS= option specifies the tick mark labels for the vertical axis, while NROWS=2 and NCOLS=2 specify a 2×2 arrangement for the tiles. The CFRAMESIDE= and CFRAMETOP= options specify fill colors for the row and column labels, and the CFILL= option specifies a fill color for the bars. The INSET statement is used to display the capability index C_{pk} for each cell. Output 10.2.1 provides evidence that both suppliers have reduced variability from 1992 to 1993.

Chapter 11
HISTOGRAM Statement

Chapter Contents

OVERVIEW 279

GETTING STARTED 279
- Creating a Histogram with Specification Limits 279
- Adding a Normal Curve to the Histogram 280
- Customizing a Histogram 284

SYNTAX 285
- Summary of Options 286
- Dictionary of Options 291

DETAILS 313
- Formulas for Fitted Curves 313
- Kernel Density Estimates 319
- Printed Output 321
- Output Data Sets 328
- ODS Tables 331
- SYMBOL and PATTERN Statement Options 332

EXAMPLES 333
- Example 11.1. Fitting a Beta Curve 333
- Example 11.2. Fitting Lognormal, Weibull, and Gamma Curves 336
- Example 11.3. Comparing Goodness-of-Fit Tests 342
- Example 11.4. Computing Capability Indices for Nonnormal Distributions . 343
- Example 11.5. Computing Kernel Density Estimates 344
- Example 11.6. Fitting a Three-Parameter Lognormal Curve 345
- Example 11.7. Annotating a Folded Normal Curve 347

Chapter 11
HISTOGRAM Statement

Overview

Histograms are typically used in process capability analysis to compare the distribution of measurements from an in-control process with its specification limits. In addition to creating histograms, you can use the HISTOGRAM statement to

- specify the midpoints or endpoints for histogram intervals
- display specification limits on histograms
- display density curves for fitted theoretical distributions (beta, exponential, gamma, Johnson S_B, Johnson S_U, lognormal, normal, and Weibull) on histograms
- request goodness-of-fit tests for fitted distributions
- display kernel density estimates on histograms
- inset summary statistics and process capability indices on histograms
- save histogram intervals and parameters of fitted distributions in output data sets
- create hanging histograms
- request graphical enhancements

Getting Started

This section introduces the HISTOGRAM statement with examples that illustrate commonly used options. Complete syntax for the HISTOGRAM statement is presented in the "Syntax" section on page 285, and advanced examples are given in the "Examples" section on page 333.

Creating a Histogram with Specification Limits

See CAPHST1 in the SAS/QC Sample Library

A semiconductor manufacturer produces printed circuit boards that are sampled to determine whether the thickness of their copper plating lies between a lower specification limit of 3.45 mils and an upper specification limit of 3.55 mils. The plating process is assumed to be in statistical control. The plating thicknesses of 100 boards are saved in a data set named TRANS, created by the following statements:

```
data trans;
   input thick @@;
   label thick='Plating Thickness (mils)';
datalines;
```

```
3.468 3.428 3.509 3.516 3.461 3.492 3.478 3.556 3.482 3.512
3.490 3.467 3.498 3.519 3.504 3.469 3.497 3.495 3.518 3.523
3.458 3.478 3.443 3.500 3.449 3.525 3.461 3.489 3.514 3.470
3.561 3.506 3.444 3.479 3.524 3.531 3.501 3.495 3.443 3.458
3.481 3.497 3.461 3.513 3.528 3.496 3.533 3.450 3.516 3.476
3.512 3.550 3.441 3.541 3.569 3.531 3.468 3.564 3.522 3.520
3.505 3.523 3.475 3.470 3.457 3.536 3.528 3.477 3.536 3.491
3.510 3.461 3.431 3.502 3.491 3.506 3.439 3.513 3.496 3.539
3.469 3.481 3.515 3.535 3.460 3.575 3.488 3.515 3.484 3.482
3.517 3.483 3.467 3.467 3.502 3.471 3.516 3.474 3.500 3.466
;
run;
```

The following statements create the histogram shown in Figure 11.1:

```
title 'Process Capability Analysis of Plating Thickness';
proc capability data=trans noprint;
   spec lsl = 3.45 llsl = 2 usl = 3.55 lusl = 2;
   histogram thick;
run;
```

A histogram is created for each variable listed after the keyword HISTOGRAM. If you specify the LINEPRINTER option in the PROC CAPABILITY statement, the histogram is displayed in line printer output, as shown in Figure 11.2. * The SPEC statement, which is optional, provides the specification limits that are displayed on the histogram. For more information on the SPEC statement, see "Syntax for the SPEC Statement" on page 183.

The NOPRINT option suppresses printed output with summary statistics for the variable THICK that would be displayed by default. See "Computing Descriptive Statistics" on page 166 for an example of this output.

Adding a Normal Curve to the Histogram

See CAPHST1 in the SAS/QC Sample Library

This example is a continuation of the preceding example.

The following statements fit a normal distribution using the thickness measurements and superimpose the fitted density curve on the histogram:

```
title 'Process Capability Analysis of Plating Thickness';
proc capability data=trans;
   spec lsl = 3.45 llsl = 2 usl = 3.55 lusl = 2;
   histogram / normal;
run;
```

The NORMAL option summarizes the fitted distribution in the printed output shown in Figure 11.3, and it specifies that the normal curve be displayed on the histogram shown in Figure 11.4.

*In Release 6.12 and previous releases of SAS/QC software, the keyword GRAPHICS was required in the PROC CAPABILITY statement to specify that the chart be created with a graphics device. In Version 7, you can specify the LINEPRINTER option to request line printer plots.

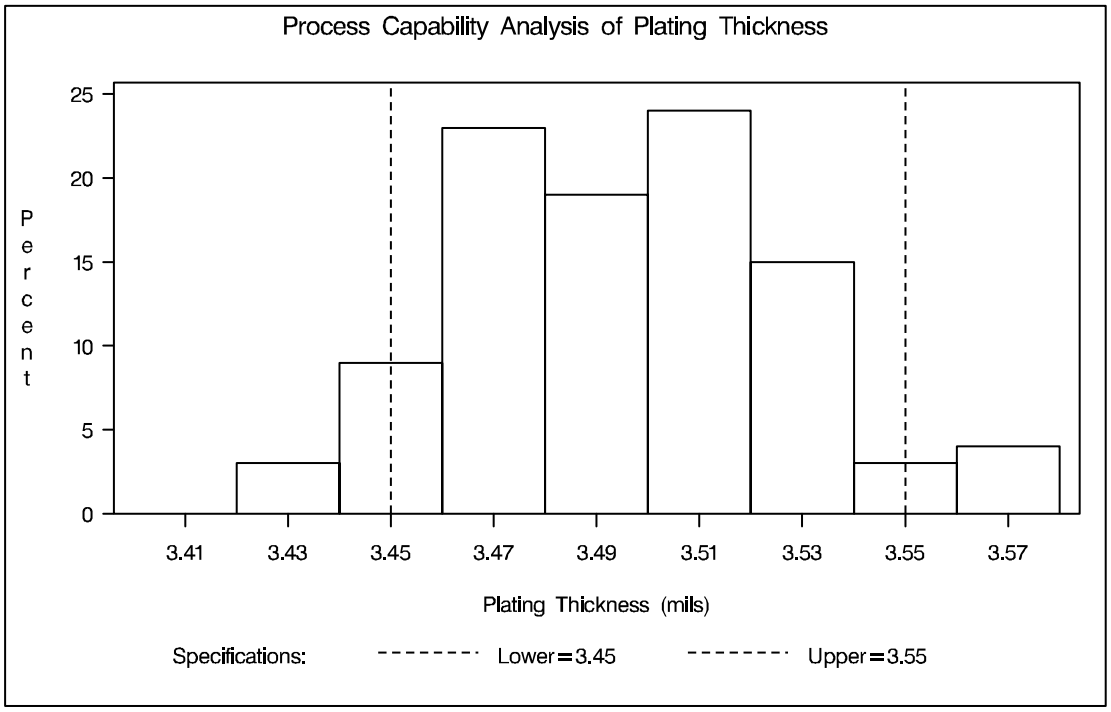


Figure 11.1. Histogram Created with Graphics Device

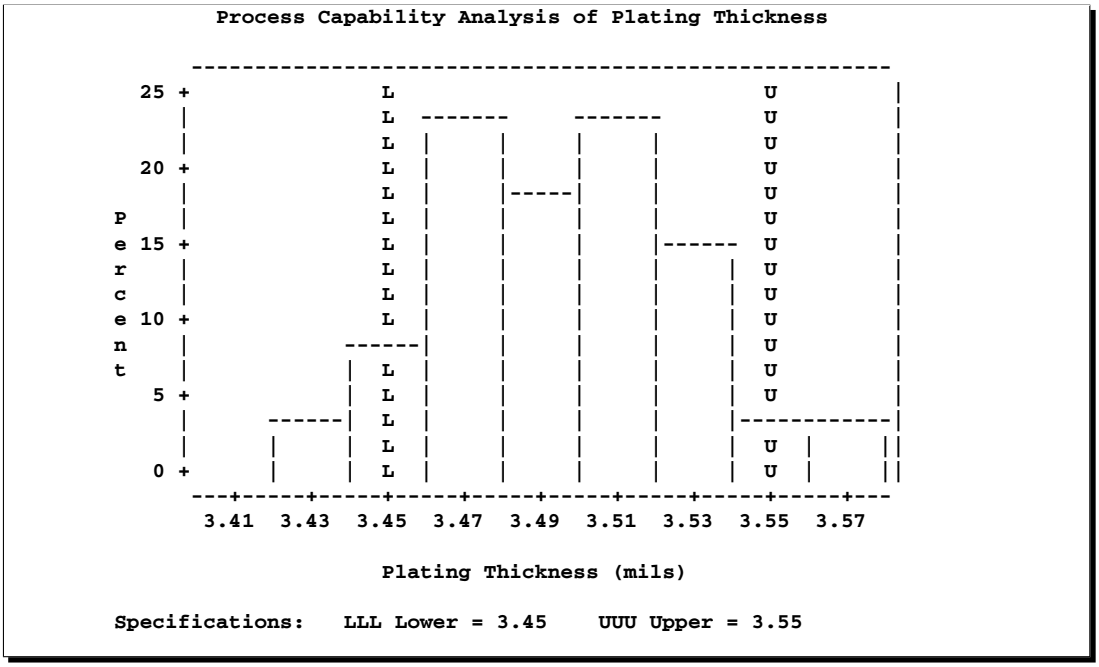


Figure 11.2. Histogram Created with Line Printer

```
            Process Capability Analysis of Plating Thickness

                        The CAPABILITY Procedure
                 Variable:  thick  (Plating Thickness (mils))

                          Specification Limits

                --------Limit--------     ------Percent-------

                Lower (LSL)  3.450000     % < LSL      8.00000
                Target                    % Between   87.00000
                Upper (USL)  3.550000     % > USL      5.00000



            Process Capability Analysis of Plating Thickness

                        The CAPABILITY Procedure
                  Fitted Normal Distribution for thick

                   Parameters for Normal Distribution

                      Parameter   Symbol   Estimate

                      Mean        Mu        3.49533
                      Std Dev     Sigma    0.032117


                   Goodness-of-Fit Tests for Normal Distribution

Test                  ----Statistic-----     DF   ------p Value------

Kolmogorov-Smirnov    D       0.05563823          Pr > D       >0.150
Cramer-von Mises      W-Sq    0.04307548          Pr > W-Sq    >0.250
Anderson-Darling      A-Sq    0.27840748          Pr > A-Sq    >0.250
Chi-Square            Chi-Sq  6.96953022      5   Pr > Chi-Sq   0.223


                     Quantiles for Normal Distribution

                              ------Quantile------
                     Percent   Observed   Estimated

                         1.0    3.42950     3.42061
                         5.0    3.44300     3.44250
                        10.0    3.45750     3.45417
                        25.0    3.46950     3.47367
                        50.0    3.49600     3.49533
                        75.0    3.51650     3.51699
                        90.0    3.53550     3.53649
                        95.0    3.55300     3.54816
                        99.0    3.57200     3.57005
```

Figure 11.3. Summary for Fitted Normal Distribution

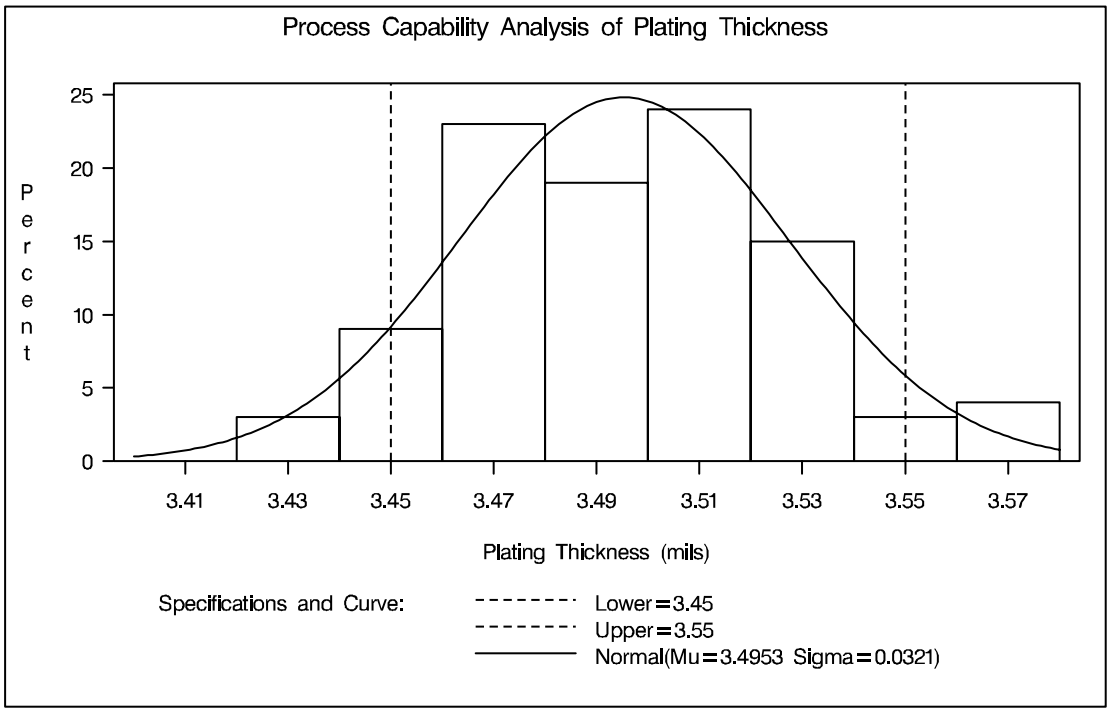


Figure 11.4. Histogram Superimposed with Normal Curve

The printed output includes the following:

- parameters for the normal curve. The normal parameters μ and σ are estimated by the sample mean ($\hat{\mu} = 3.49533$) and the sample standard deviation ($\hat{\sigma} = 0.03211691$).
- a chi-square goodness-of-fit test. Compared to the usual cutoff values of 0.05 and 0.10, the p-value of 0.2229 for this test indicates that the thicknesses are normally distributed.
- goodness-of-fit tests based on the empirical distribution function (EDF): the Anderson-Darling, Cramer-von Mises, and Kolmogorov-Smirnov tests. The p-values for these tests are smaller than the usual cutoff values of 0.05 and 0.10, indicating that the thicknesses are normally distributed.
- a chi-square goodness-of-fit test. The p-value of 0.2229 for this test indicates that the thicknesses are normally distributed. In general EDF tests (when available) are preferable to chi-square tests. See the "EDF Goodness-of-Fit Tests" section on page 323 for details.
- observed and estimated percentages outside the specification limits
- observed and estimated quantiles

For details, including formulas for the goodness-of-fit tests, see "Printed Output" on page 321. Note that the NOPRINT option in the PROC CAPABILITY statement suppresses only the printed output with summary statistics for the variable THICK.

To suppress the printed output in Figure 11.3, specify the NOPRINT option enclosed in parentheses after the NORMAL option, as on page 284.

The NORMAL option is one of many options that you can specify in the HISTOGRAM statement. See the "Syntax" section on page 285 for a complete list of options or the "Dictionary of Options" section on page 291 for detailed descriptions of options.

Customizing a Histogram

See CAPHST1 in the SAS/QC Sample Library

This example is a continuation of the preceding example. The following statements show how you can use HISTOGRAM statement options and INSET statements to customize a histogram:

```
title 'Process Capability Analysis of Plating Thickness';
proc capability data=trans noprint;
   spec lsl = 3.45 llsl = 2 usl = 3.55 lusl = 3;
   histogram thick / normal(color=yellow w=3)
                     midpoints = 3.4 to 3.6 by 0.025
                     vscale    = count
                     cfill     = yellow
                     nospeclegend ;
   inset lsl usl    / cfill = blank;
   inset n mean (5.2) cpk (5.2) / cfill = blank;
run;
```

The histogram is displayed in Figure 11.5.

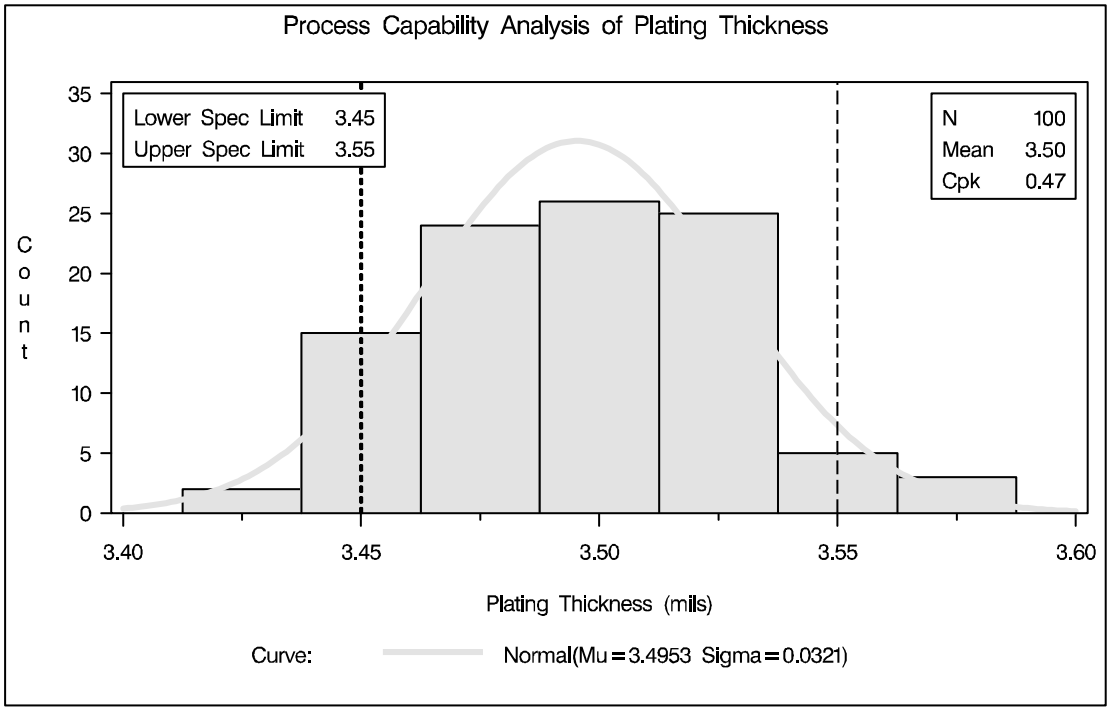


Figure 11.5. Customizing the Appearance of the Histogram

The MIDPOINTS= option specifies a list of values to use as bin midpoints. The VSCALE=COUNT option requests a vertical axis scaled in counts rather than percents. The CFILL= option specifies a color for the histogram bars. The INSET statements inset the specification limits and summary statistics. The NOSPECLEGEND option suppress the default legend for the specification limits that is shown in Figure 11.4.

For more information about HISTOGRAM statement options, see "Dictionary of Options" on page 291. For details on the INSET statement, see Chapter 12, "INSET Statement," on page 353.

Syntax

The syntax for the HISTOGRAM statement is as follows:

HISTOGRAM <*variables* > < **/** *options* >**;**

You can specify the keyword HIST as an alias for HISTOGRAM. You can use any number of HISTOGRAM statements after a PROC CAPABILITY statement. The components of the HISTOGRAM statement are described as follows.

variables
are the process variables for which histograms are to be created. If you specify a VAR statement, the *variables* must also be listed in the VAR statement. Otherwise, the *variables* can be any numeric variables in the input data set. If you do not specify *variables* in a VAR statement or in the HISTOGRAM statement, then by default, a histogram is created for each numeric variable in the DATA= data set. If you use a VAR statement and do not specify any *variables* in the HISTOGRAM statement, then by default, a histogram is created for each variable listed in the VAR statement.

For example, suppose a data set named STEEL contains exactly two numeric variables named LENGTH and WIDTH. The following statements create two histograms, one for LENGTH and one for WIDTH:

```
proc capability data=steel;
   histogram;
run;
```

Likewise, the following statements create histograms for LENGTH and WIDTH:

```
proc capability data=steel;
   var length width;
   histogram;
run;
```

The following statements create a histogram for LENGTH only:

```
proc capability data=steel;
   var length width;
   histogram length;
run;
```

options

add features to the histogram. Specify all *options* after the slash (/) in the HISTOGRAM statement.

For example, in the following statements, the NORMAL option displays a fitted normal curve on the histogram, the MIDPOINTS= option specifies midpoints for the histogram, and the CTEXT= option specifies the color of the text:

```
proc capability data=steel;
   histogram length / normal
                      midpoints = 5.6 5.8 6.0 6.2 6.4
                      ctext     = yellow;
run;
```

Summary of Options

The following tables list the HISTOGRAM statement *options* by function. For detailed descriptions, see "Dictionary of Options" on page 291.

Parametric Density Estimation Options

Table 11.1 lists options that display a parametric density estimate on the histogram.

Table 11.1. Parametric Distribution Options

BETA(*beta-options*)	fits beta distribution with threshold parameter θ, scale parameter σ, and shape parameters α and β
EXPONENTIAL(*exponential-options*)	fits exponential distribution with threshold parameter θ and scale parameter σ
GAMMA(*gamma-options*)	fits gamma distribution with threshold parameter θ, scale parameter σ, and shape parameter α
LOGNORMAL(*lognormal-options*)	fits lognormal distribution with threshold parameter θ, scale parameter ζ, and shape parameter σ
NORMAL(*normal-options*)	fits normal distribution with mean μ and standard deviation σ
SB(*SB-options*)	fits Johnson S_B distribution with threshold parameter θ, scale parameter σ, and shape parameters δ and γ
SU(*SU-options*)	fits Johnson S_U distribution with location parameter θ, scale parameter σ, and shape parameters δ and γ
WEIBULL(*Weibull-options*)	fits Weibull distribution with threshold parameter θ, scale parameter σ, and shape parameter c

Table 11.2 through Table 11.10 list options that specify parameters for fitted parametric distributions and that control the display of fitted curves. Specify these options in parentheses after the distribution keyword. For example, the following statements fit a normal curve with the keyword NORMAL:

```
proc capability;
   histogram / normal(color=red mu=10 sigma=0.5);
run;
```

The COLOR= *normal-option* draws the curve in red, and the MU= and SIGMA= *normal-options* specify the parameters $\mu = 10$ and $\sigma = 0.5$ for the curve. Note that the sample mean and sample standard deviation are used to estimate μ and σ, respectively, when the MU= and SIGMA= options are not specified.

Table 11.2. Options Used with All Parametric Distribution Options

COLOR=*color*	specifies color of fitted density curve
FILL	fills area under fitted density curve
INDICES	calculates capability indices based on fitted distribution
L=*linetype*	specifies line type of fitted curve
MIDPERCENTS	prints table of midpoints of histogram intervals
NOPRINT	suppresses printed output summarizing fitted curve
PERCENTS=*value-list*	lists percents for which quantiles calculated from data and quantiles estimated from fitted curve are tabulated
SYMBOL='*character*'	specifies character used to plot fitted density curve if histogram is produced on a line printer
W=n	specifies width of fitted density curve

Table 11.3. Beta-Options

ALPHA=*value*	specifies first shape parameter α for fitted beta curve
BETA=*value*	specifies second shape parameter β for fitted beta curve
SIGMA=*value*|EST	specifies scale parameter σ for fitted beta curve
THETA=*value*|EST	specifies lower threshold parameter θ for fitted beta curve

Table 11.4. Exponential-Options

SIGMA=*value*	specifies scale parameter σ for fitted exponential curve
THETA=*value*|EST	specifies threshold parameter θ for fitted exponential curve

Table 11.5. Gamma-Options

ALPHADELTA=*value*	specifies change in successive estimates of α at which the Newton-Raphson approximation of $\hat{\alpha}$ terminates
ALPHAINITIAL=*value*	specifies initial value for α in Newton-Raphson approximation of $\hat{\alpha}$
MAXITER=n	specifies maximum number of iterations in Newton-Raphson approximation of $\hat{\alpha}$
SIGMA=*value*	specifies scale parameter σ for fitted gamma curve
ALPHA=*value*	specifies shape parameter α for fitted gamma curve
THETA=*value*|EST	specifies threshold parameter θ for fitted gamma curve

Table 11.6. Lognormal-Options

ZETA=*value*	specifies scale parameter ζ for fitted lognormal curve
SIGMA=*value*	specifies shape parameter σ for fitted lognormal curve
THETA=*value*|EST	specifies threshold parameter θ for fitted lognormal curve

Table 11.7. Normal-Options

MU=*value*	specifies mean μ for fitted normal curve
SIGMA=*value*	specifies standard deviation σ for fitted normal curve

Table 11.8. S_B-Options

DELTA=*value*	specifies first shape parameter δ for fitted S_B curve
FITINTERVAL=*value*	specifies z-value for method of percentiles
FITMETHOD=MLE| PERCENTILE| MOMENTS	specifies method of parameter estimation
GAMMA=*value*	specifies second shape parameter γ for fitted S_B curve
SIGMA=*value*|EST	specifies scale parameter σ for fitted S_B curve
THETA=*value*|EST	specifies lower threshold parameter θ for fitted S_B curve
FITTOLERANCE=*value*	specifies tolerance for method of percentiles

Table 11.9. S_U-Options

DELTA=*value*	specifies first shape parameter δ for fitted S_U curve
FITINTERVAL=*value*	specifies z-value for method of percentiles
FITMETHOD=MLE| PERCENTILE| MOMENTS	specifies method of parameter estimation
GAMMA=*value*	specifies second shape parameter γ for fitted S_U curve
SIGMA=*value*|EST	specifies scale parameter σ for fitted S_U curve
THETA=*value*|EST	specifies location parameter θ for fitted S_U curve
FITTOLERANCE=*value*	specifies tolerance for method of percentiles

Table 11.10. Weibull-Options

C=*value*	specifies shape parameter c for fitted Weibull curve
CDELTA=*value*	specifies change in successive estimates of c at which the Newton-Raphson approximation of $\hat{c}$ terminates
CINITIAL=*value*	specifies initial value for c in Newton-Raphson approximation of $\hat{c}$
MAXITER=n	specifies maximum number of iterations in Newton-Raphson approximation of $\hat{c}$
SIGMA=*value*	specifies scale parameter σ for fitted Weibull curve
THETA=*value*|EST	specifies threshold parameter θ for fitted Weibull curve

Nonparametric Density Estimation Options

Table 11.11. Kernel Density Estimation Options

KERNEL(*kernel-options*)	fits kernel density estimates

Specify the options listed in Table 11.12 in parentheses after the keyword KERNEL to control features of kernel density estimates requested with the KERNEL option.

Table 11.12. Kernel-Options

C=*value* | MISE	specifies standardized bandwidth parameter c for fitted kernel density estimate
COLOR=*color*	specifies color of the fitted kernel density curve
FILL	fills area under fitted kernel density curve
K=NORMAL | QUADRATIC | TRIANGULAR	specifies type of kernel function
L=*linetype*	specifies line type used for fitted kernel density curve
LOWER=	specifies lower bound for fitted kernel density curve
SYMBOL='*character*'	specifies character used to plot fitted kernel density curve if the histogram is produced on a line printer
UPPER=	specifies upper bound for fitted kernel density curve
W=n	specifies line width for fitted kernel density curve

General Options

Table 11.13 through Table 11.16 summarize general options for the HISTOGRAM statement, including options for enhancing charts and producing output data sets.

Table 11.13. General Histogram Layout Options

CURVELEGEND=*name* | NONE	specifies LEGEND statement for curves
ENDPOINTS=*value-list*	lists endpoints for histogram intervals
FORCEHIST	forces creation of histogram
HANGING	constructs hanging histogram
HREF=*value-list*	specifies reference lines perpendicular to the horizontal axis
HREFLABELS='*label1*' . . . '*labeln*'	specifies labels for HREF= lines
MIDPERCENTS	prints table of histogram intervals
MIDPOINTS=*value-list*	lists midpoints for histogram intervals
NENDPOINTS=n	specifies number of histogram interval endpoints
NMIDPOINTS=n	specifies number of histogram interval midpoints
NOBARS	suppresses histogram bars
NOCURVELEGEND	suppresses legend for curves
NOFRAME	suppresses frame around plotting area
NOLEGEND	suppresses legend
NOPLOT	suppresses plot
NOSPECLEGEND	suppresses specifications legend
RTINCLUDE	includes right endpoint in interval
SPECLEGEND=*name* | NONE	specifies LEGEND statement for specification limits
VREF=*value-list*	specifies reference lines perpendicular to the vertical axis
VREFLABELS='*label1*' . . . '*labeln*'	specifies labels for VREF= lines
VSCALE=COUNT | PERCENT | PROPORTION	specifies scale for vertical axis

Table 11.14. Options to Create Output Data Sets

OUTFIT=*SAS-data-set*	specifies information on fitted curves
OUTHISTOGRAM=*SAS-data-set*	specifies information on histogram intervals

Table 11.15. Options to Enhance Histograms Produced on Line Printers

HREFCHAR='*character*'	specifies line character for HREF= lines
VREFCHAR='*character*'	specifies line character for VREF= lines

Table 11.16. Options to Enhance Histograms Produced on Graphics Devices

ANNOTATE=*SAS-data-set*	specifies annotate data set
BARLABEL=COUNT | PERCENT | PROPORTION	produces labels above histogram bars
BMCBOXFILL=*color*	specifies fill color for box-and-whisker plot in bottom margin
BMCFRAME=*color*	specifies fill color bottom margin plot frame
BMCOLOR=*color*	specifies color for bottom margin plot
BMMARGIN=*height*	specifies height of margin for bottom margin plot
BMPLOT=*keyword*	requests a plot in bottom margin of histogram
CAXIS=*color*	specifies color for axis
CBARLINE=*color*	specifies color of outlines of histogram bars
CFILL=*color*	specifies color for filling under curve
CFRAME=*color*	specifies color for frame
CHREF=*color*	specifies color for HREF= lines
CLIPSPEC=CLIP | NOFILL	clips histogram bars at specification limits if there are no observations beyond the limits
CTEXT=*color*	specifies color for text
CVREF=*color*	specifies color for VREF= lines
DESCRIPTION='*string*'	specifies description for plot in graphics catalog
FONT=*font*	specifies software font for text
FRONTREF	draws reference lines in front of histogram bars
HAXIS=*name*	specifies AXIS statement for horizontal axis
HMINOR=n	specifies number of horizontal minor tick marks
LEGEND=*name* | NONE	identifies LEGEND statement
LHREF=*linetype*	specifies line style for HREF= lines
LVREF=*linetype*	specifies line style for VREF= lines
MIDPTAXIS=*name*	specifies name of AXIS statement for horizontal axis
NAME='*string*'	specifies name for plot in graphics catalog
PCTAXIS=*name*|*value-list*	specifies AXIS statement or values for vertical axis
PFILL=*pattern*	specifies pattern for filling under curve
VAXIS=*name*|*value-list*	specifies AXIS statement or values for vertical axis
VMINOR=n	specifies number of vertical minor tick marks
WBARLINE=n	specifies line thickness for bar outlines

Dictionary of Options

The following entries provide detailed descriptions of options for the HISTOGRAM statement. The marginal notes *Graphics* and *Line Printer* identify options that can be used only with graphics devices and line printers, respectively.

ALPHA=*value*

specifies the shape parameter α for fitted curves requested with the BETA and GAMMA options. Enclose the ALPHA= option in parentheses after the BETA or GAMMA options. If you do not specify a value for α, the procedure calculates a maximum likelihood estimate. See Example 11.1 on page 333. You can specify A= as an alias for ALPHA= if you use it as a *beta-option*. You can specify SHAPE= as an alias for ALPHA= if you use it as a *gamma-option*.

ALPHADELTA=*value*

specifies the change in successive estimates of $\hat{\alpha}$ at which iteration terminates in the Newton-Raphson approximation of the maximum likelihood estimate of α for curves requested by the GAMMA option. Enclose the ALPHADELTA= option in parentheses after the GAMMA option. Iteration continues until the change in α is less than the value specified or until the number of iterations exceeds the value of the MAXITER= option (see page 303). The default value is 0.00001.

ALPHAINITIAL=*value*

specifies the initial value for $\hat{\alpha}$ in the Newton-Raphson approximation of the maximum likelihood estimate of α for fitted gamma distributions requested with the GAMMA option. Enclose the ALPHAINITIAL= option in parentheses after the GAMMA option. The default value is Thom's approximation of the estimate of α. Refer to Johnson *et al.* (1994).

ANNOTATE=*SAS-data-set*
ANNO=*SAS-data-set*

Graphics

specifies an input data set containing annotate variables as described in *SAS/GRAPH Software: Reference*. See Example 11.7 on page 347. The ANNOTATE= data set you specify in the HISTOGRAM statement is used for all plots created by the statement. You can also specify an ANNOTATE= data set in the PROC CAPABILITY statement to enhance all plots created by the procedure; for more information, see "ANNOTATE= Data Sets" on page 189.

BARLABEL=COUNT | PERCENT | PROPORTION

Graphics

displays labels above the histogram bars. If you specify BARLABEL=COUNT, the label shows the number of observations associated with a given bar. BARLABEL=PERCENT shows the percent of observations represented by that bar. If you specify BARLABEL=PROPORTION, the label displays the proportion of observations associated with the bar.

BETA<(*beta-options* **)>**

displays a fitted beta density curve on the histogram. The curve equation is

$$p(x) = \begin{cases} \frac{(x-\theta)^{\alpha-1}(\sigma+\theta-x)^{\beta-1}}{B(\alpha,\beta)\sigma^{(\alpha+\beta-1)}} h \times 100\% & \text{for } \theta < x < \theta+\sigma \\ 0 & \text{for } x \le \theta \text{ or } x \ge \theta+\sigma \end{cases}$$

where $B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$ and

θ = lower threshold parameter (lower endpoint parameter)
σ = scale parameter ($\sigma > 0$)
α = shape parameter ($\alpha > 0$)
β = shape parameter ($\beta > 0$)
h = width of histogram interval

The beta distribution is bounded below by the parameter θ and above by the value $\theta + \sigma$. You can specify θ and σ using the THETA= and SIGMA= *beta-options*. The following statements fit a beta distribution bounded between 50 and 75, using maximum likelihood estimates for α and β:

```
proc capability;
   histogram length / beta(theta=50 sigma=25);
run;
```

In general, the default values for THETA= and SIGMA= are 0 and 1, respectively. You can specify THETA=EST and SIGMA=EST to request maximum likelihood estimates for θ and σ.

The beta distribution has two shape parameters, α and β. If these parameters are known, you can specify their values with the ALPHA= and BETA= *beta-options*. If you do not specify values, the procedure calculates maximum likelihood estimates for α and β.

The BETA option can appear only once in a HISTOGRAM statement. Table 11.2 (page 287) and Table 11.3 (page 287) list options you can specify with the BETA option. See Example 11.1 on page 333. Also see "Formulas for Fitted Curves" on page 313.

BETA=*value*
B=*value*
specifies the second shape parameter β for beta density curves requested with the BETA option. Enclose the BETA= option in parentheses after the BETA option. If you do not specify a value for β, the procedure calculates a maximum likelihood estimate. See Example 11.1 on page 333.

BMCBOXFILL=*color*
specifies the fill color for a box-and-whisker plot in a bottom margin requested with the BMPLOT= option. By default, the box-and-whisker plot is not filled. *Graphics*

BMCFRAME=*color*
specifies the color for filling the frame of a bottom margin plot requested with the BMPLOT= option. By default, this area is not filled. *Graphics*

BMCOLOR=*color*
specifies the color of a carpet plot or dot plot, or the outline color of a box-and-whisker plot, in a bottom margin plot requested with the BMPLOT= option. *Graphics*

BMMARGIN=*height*

Graphics

specifies the height in screen percentage units of a bottom margin plot requested with the BMPLOT= option. By default, a bottom margin plot occupies 15 percent of the vertical display space.

BMPLOT=CARPET | DOTPLOT | SKELETAL | SCHEMATIC

Graphics

produces a carpet plot, dot plot, or box-and-whisker plot along the bottom margin of a histogram. A carpet plot or dot plot shows the distribution of individual observations along the histogram's horizontal axis. A carpet plot represents each observation with a vertical line. A dot plot marks each observation with a symbol. A box-and-whisker plot gives a summary of the data distribution that a histogram alone does not provide. The left and right edges of the box are located at the first and third quartiles. A central vertical line is drawn at the median and a symbol is plotted inside the box at the mean. If you specify the SKELETAL keyword, a box-and-whisker plot is produced with whiskers extending to the minimum and maximum values. If you specify SCHEMATIC, a *schematic* box-and-whisker plot is produced. In a schematic box-and-whisker plot, the whiskers extend to the smallest value within the *lower fence* and the largest value within the *upper fence*. Fences are defined in terms of the interquartile range (IQR). The lower fence is 1.5 IQR below the first quartile and the upper fence is 1.5 IQR above the third quartile. Each observation outside the fences is plotted with a symbol.

C=*value*

specifies the shape parameter c for Weibull density curves requested with the WEIBULL option. Enclose the C= option in parentheses after the WEIBULL option. If you do not specify a value for c, the procedure calculates a maximum likelihood estimate. See Example 11.2 on page 336. You can specify the SHAPE= option as an alias for the C= option.

C=*value-list* **| MISE**

specifies the standardized bandwidth parameter c for kernel density estimates requested with the KERNEL option. Enclose the C= option in parentheses after the KERNEL option. You can specify up to five values to request multiple estimates. You can also specify the C=MISE option, which produces the estimate with a bandwidth that minimizes the approximate mean integrated square error (MISE). For example, the following statements compute three density estimates:

```
proc capability;
   histogram length / kernel(c=0.5 1.0 mise);
run;
```

The first two estimates have standardized bandwidths of 0.5 and 1.0, respectively, and the third has a bandwidth that minimizes the approximate MISE.

You can also use the C= option with the K= option, which specifies the kernel function, to compute multiple estimates. If you specify more kernel functions than bandwidths, the last bandwidth in the list is repeated for the remaining estimates. Likewise, if you specify more bandwidths than kernel functions, the last kernel function is repeated for the remaining estimates. For example, the following statements compute three density estimates:

```
proc capability;
   histogram length / kernel(c=1 2 3 k=normal quadratic);
run;
```

The first uses a normal kernel and a bandwidth of 1, the second uses a quadratic kernel and a bandwidth of 2, and the third uses a quadratic kernel and a bandwidth of 3. See Example 11.5 on page 344.

If you do not specify a value for c, the bandwidth that minimizes the approximate MISE is used for all the estimates.

CAXIS=*color*
CAXES=*color*

specifies the color used for the axes and tick marks. This option overrides any COLOR= specifications in an AXIS statement. The default is the first color in the device color list. *Graphics*

CBARLINE=*color*

specifies the color of the outline of histogram bars. This option overrides the C= option in the SYMBOL1 statement. The default is the first color in the device color list. *Graphics*

CDELTA=*value*

specifies the change in successive estimates of c at which iterations terminate in the Newton-Raphson approximation of the maximum likelihood estimate of c for fitted Weibull curves requested by the WEIBULL option. Enclose the CDELTA= option in parentheses after the WEIBULL option. Iteration continues until the change in c between consecutive steps is less than the value specified or until the number of iterations exceeds the value of the MAXITER= option (see page 303). The default value is 0.00001. For examples, see the entry for the WEIBULL option.

CFILL=*color*

specifies a color used to fill the bars of the histogram (or the area under a fitted curve if you also specify the FILL option). See the entries for the FILL and PFILL= options for additional details. See Figure 11.5 on page 284 and Output 11.1.1 on page 335. Refer to *SAS/GRAPH Software: Reference* for a list of colors. By default, bars and curve areas are not filled. *Graphics*

CFRAME=*color*
CFR=*color*

specifies the color for the area enclosed by the axes and frame. The area is not filled by default. *Graphics*

CHREF=*color*
CH=*color*

specifies the color for horizontal axis reference lines requested by the HREF= option. The default is the first color in the device color list. *Graphics*

CINITIAL=*value*

specifies the initial value for $\hat{c}$ in the Newton-Raphson approximation of the maximum likelihood estimate of c for Weibull curves requested with the WEIBULL option. Enclose the CINITIAL= option in parentheses after the WEIBULL option. The default value is 1.8 (refer to Johnson *et al.* 1994).

CLIPSPEC=CLIP | NOFILL

Graphics

specifies that histogram bars are clipped at the upper and lower specification limit lines when there are no observations outside the specification limits. The bar intersecting the lower specification limit is clipped if there are no observations less than the lower limit; the bar intersecting the upper specification limit is clipped if there are no observations greater than the upper limit. If you specify CLIPSPEC=CLIP, the histogram bar is truncated at the specification limit. If you specify CLIPSPEC=NOFILL, the portion of a filled histogram bar outside the specification limit is left unfilled. Specifying CLIPSPEC=NOFILL when histogram bars are not filled has no effect.

COLOR=*color*

Graphics

specifies the color of the density curve. Enclose the COLOR= option in parentheses after the distribution option or the KERNEL option. See Example 11.1 on page 333. If you use the COLOR= option with the KERNEL option, you can specify a list of up to five colors in parentheses for multiple kernel density estimates. If there are more estimates than colors, the last color specified is used for the remaining estimates.

CTEXT=*color*

Graphics

specifies the color for tick mark values and axis labels. The default is the color specified for the CTEXT= option in the GOPTIONS statement. In the absence of a GOPTIONS statement, the default color is the first color in the device color list.

CURVELEGEND=*name* **| NONE**

specifies the name of a LEGEND statement describing the legend for specification limits and fitted curves. Specifying CURVELEGEND=NONE suppresses the legend for fitted curves; this is equivalent to specifying the NOCURVELEGEND option.

CVREF=*color*

CV=*color*

Graphics

specifies the color for lines requested with the VREF= option. The default is the first color in the device color list.

DELTA=*value*

specifies the first shape parameter δ for Johnson S_B and Johnson S_U density curves requested with the SB and SU options. Enclose the DELTA= option in parentheses after the SB or SU option. If you do not specify a value for δ, the procedure calculates an estimate.

DESCRIPTION='*string***'**

DES='*string***'**

Graphics

specifies a description, up to 40 characters, that appears in the PROC GREPLAY master menu. The default is the variable name.

ENDPOINTS
ENDPOINTS=*value-list*

specifies that histogram interval endpoints, rather than midpoints, are aligned with horizontal axis tick marks. If you specify ENDPOINTS, the number of histogram intervals is based on the number of observations using the method of Terrell and Scott (1985). If you specify ENDPOINTS=*value-list*, the *values* must be listed in increasing order and must be evenly spaced. All observations in the input data set, as well as any specification limits, must lie between the first and last values specified. The same *value-list* is used for all variables.

EXPONENTIAL<**(***exponential-options* **)**>
EXP<**(***exponential-options* **)**>

displays a fitted exponential density curve on the histogram. The curve equation is

$$p(x) = \begin{cases} \frac{h \times 100\%}{\sigma} \exp(-(\frac{x-\theta}{\sigma})) & \text{for } x \geq \theta \\ 0 & \text{for } x < \theta \end{cases}$$

where

$\theta =$ threshold parameter
$\sigma =$ scale parameter $(\sigma > 0)$
$h =$ width of histogram interval

The parameter θ must be less than or equal to the minimum data value. You can specify θ with the THETA= *exponential-option*. The default value for θ is zero. If you specify THETA=EST, a maximum likelihood estimate is computed for θ. You can specify σ with the SIGMA= *exponential-option*. By default, a maximum likelihood estimate is computed for σ. For example, the following statements fit an exponential curve with $\theta = 10$ and with a maximum likelihood estimate for σ:

```
proc capability;
   histogram / exponential(theta=10 l=2 color=red);
run;
```

The curve is red and has a line type of 2. The EXPONENTIAL option can appear only once in a HISTOGRAM statement. Table 11.2 (page 287) and Table 11.4 (page 287) list options you can specify with the EXPONENTIAL option. See "Formulas for Fitted Curves" on page 313.

FILL

Graphics

fills areas under a parametric density curve or kernel density estimate with colors and patterns. Enclose the FILL option in parentheses after a curve option or the KERNEL option, as in the following statements:

```
proc capability;
   histogram length / normal(fill) cfill=green pfill=solid;
run;
```

Depending on the area to be filled (outside or between the specification limits), you can specify the color and pattern with options in the SPEC statement and HISTOGRAM statement, as summarized in the following table:

Area Under Curve	Statement	Option
between specification limits	HISTOGRAM HISTOGRAM	CFILL=*color* PFILL=*pattern*
left of lower specification limit	SPEC SPEC	CLEFT=*color* PLEFT=*pattern*
right of upper specification limit	SPEC SPEC	CRIGHT=*color* PRIGHT=*pattern*

If you do not display specification limits, the CFILL= and PFILL= options specify the color and pattern for the entire area under the curve. Solid fills are used by default if patterns are not specified. You can specify the FILL option with only one fitted curve. For an example, see Output 11.1.1 on page 335. Refer to *SAS/GRAPH Software: Reference* for a list of available patterns and colors. If you do not specify the FILL option but specify the options in the preceding table, the colors and patterns are applied to the corresponding areas under the histogram.

FITINTERVAL=*value*

specifies the value of z for the method of percentiles when this method is used to fit a Johnson S_B or Johnson S_U distribution. The FITINTERVAL= option is specified in parentheses after the SB or SU option. The default *value* of z is 0.524.

FITMETHOD=PERCENTILE|MLE|MOMENTS

specifies the method used to estimate the parameters of a Johnson S_B or Johnson S_U distribution. The FITMETHOD= option is specified in parentheses after the SB or SU option. By default, the method of percentiles is used.

FITTOLERANCE=*value*

specifies the tolerance value for the ratio criterion when the method of percentiles is used to fit a Johnson S_B or Johnson S_U distribution. The FITTOLERANCE= option is specified in parentheses after the SB or SU option. The default *value* is 0.01.

FONT=*font*

Graphics

specifies a software font for reference line and axis labels. You can also specify fonts for axis labels in an AXIS statement. The FONT= font takes precedence over the FTEXT= font specified in the GOPTIONS statement. Hardware characters are used by default.

FORCEHIST

forces the creation of a histogram if there is only one unique observation. By default, a histogram is not created if the standard deviation of the data is zero.

FRONTREF

Graphics

draws reference lines requested with the HREF= and VREF= options in front of the histogram bars. By default, reference lines are drawn behind the histogram bars and can be obscured by them.

GAMMA<**(***gamma-options***)**>

displays a fitted gamma density curve on the histogram. The curve equation is

$$p(x) = \begin{cases} \frac{h \times 100\%}{\Gamma(\alpha)\sigma}(\frac{x-\theta}{\sigma})^{\alpha-1}\exp(-(\frac{x-\theta}{\sigma})) & \text{for } x > \theta \\ 0 & \text{for } x \leq \theta \end{cases}$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)
α = shape parameter ($\alpha > 0$)
h = width of histogram interval

The parameter θ for the gamma distribution must be less than the minimum data value. You can specify θ with the THETA= *gamma-option*. The default value for θ is 0. If you specify THETA=EST, a maximum likelihood estimate is computed for θ. In addition, the gamma distribution has a shape parameter α and a scale parameter σ. You can specify these parameters with the ALPHA= and SIGMA= *gamma-options*. By default, maximum likelihood estimates are computed for α and σ. For example, the following statements fit a gamma curve with $\theta = 4$ and with maximum likelihood estimates for α and σ:

```
proc capability;
   histogram length / gamma(theta=4);
run;
```

Note that the maximum likelihood estimate of α is calculated iteratively using the Newton-Raphson approximation. The ALPHADELTA=, ALPHAINITIAL=, and MAXITER= *gamma-options* control the approximation.

The GAMMA option can appear only once in a HISTOGRAM statement. Table 11.2 (page 287) and Table 11.5 (page 287) list the options you can specify with the GAMMA option. See Example 11.2 on page 336 and "Formulas for Fitted Curves" on page 313.

GAMMA=*value*

specifies the second shape parameter γ for Johnson S_B and Johnson S_U density curves requested with the SB and SU options. Enclose the GAMMA= option in parentheses after the SB or SU option. If you do not specify a value for γ, the procedure calculates an estimate.

HANGING
HANG

requests a hanging histogram , as illustrated in Figure 11.6.

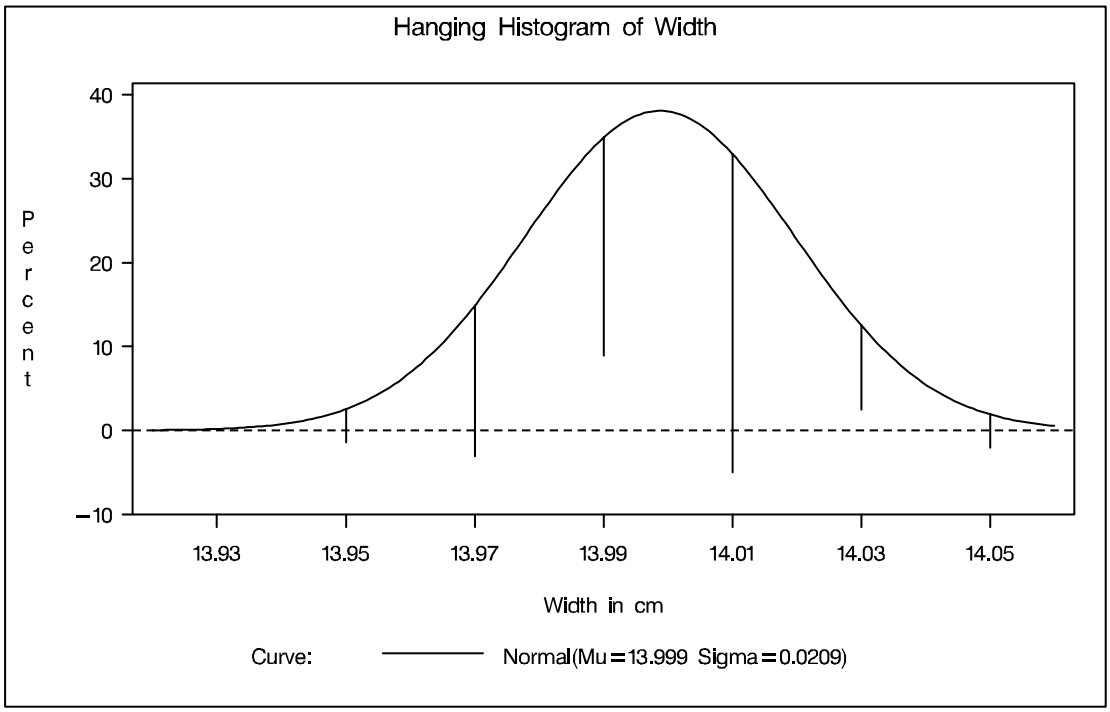


Figure 11.6. Hanging Histogram

You can use the HANGING option with only one fitted density curve. A hanging histogram aligns the tops of the histogram bars (displayed as lines) with the fitted curve. The lines are positioned at the midpoints of the histogram bins. A hanging histogram is a goodness-of-fit diagnostic in the sense that the closer the lines are to the horizontal axis, the better the fit. Hanging histograms are discussed by Tukey (1977), Wainer (1974), and Velleman and Hoaglin (1981).

HAXIS=*name*

Graphics

specifies the name of an AXIS statement describing the horizontal axis. You can specify the MIDPTAXIS= option as an alias for the HAXIS= option. See the entry for the MIDPOINTS= option for a syntax example.

HMINOR=*n*
HM=*n*

Graphics

specifies the number of minor tick marks between each major tick mark on the horizontal axis. Minor tick marks are not labeled. The default is 0.

HREF=*value-list*

draws reference lines perpendicular to the horizontal axis at the values specified. See Output 11.1.1 on page 335. Also see the CHREF=, HREFCHAR=, and LHREF= options.

HREFCHAR='*character***'**
specifies the character used to form the lines requested by the HREF= option. The default is the vertical bar (|). *Line Printer*

HREFLABELS='*label1' . . . 'labeln'*
HREFLABEL='*label1' . . . 'labeln'*
HREFLAB='*label1' . . . 'labeln'*
specifies labels for the lines requested by the HREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can have up to 16 characters. See Output 11.1.1 on page 335.

INDICES
requests capability indices based on the fitted distribution. Enclose the keyword INDICES in parentheses after the distribution keyword. See "Indices Using Fitted Curves" on page 325 for computational details and see Output 11.4.2 on page 344.

K=NORMAL | QUADRATIC | TRIANGULAR
specifies the kernel function (normal, quadratic, or triangular) used to compute a kernel density estimate. Enclose the K= option in parentheses after the KERNEL option, as in the following statements:

```
proc capability;
   histogram length / kernel(k=quadratic);
run;
```

You can specify kernel functions for up to five estimates. You can also use the K= option together with the C= option, which specifies standardized bandwidths. If you specify more kernel functions than bandwidths, the last bandwidth in the list is repeated for the remaining estimates. Likewise, if you specify more bandwidths than kernel functions, the last kernel function is repeated for the remaining estimates. For example, the following statements compute three estimates with bandwidths of 0.5, 1.0, and 1.5:

```
proc capability;
   histogram length / kernel(c=0.5 1.0 1.5 k=normal quadratic);
run;
```

The first estimate uses a normal kernel, and the last two estimates use a quadratic kernel. By default, a normal kernel is used.

KERNEL<(*kernel-options* **)>**
superimposes up to five kernel density estimates on the histogram. You can specify the *kernel-options* described in the following table:

FILL	specifies that the area under the curve is to be filled
COLOR=	specifies the color of the curve
L=	specifies the line style for the curve
LOWER=	specifies the lower bound for the curve
UPPER=	specifies the upper bound for the curve
W=	specifies the width of the curve
K=	specifies the type of kernel function
C=	specifies the smoothing parameter
SYMBOL=	specifies the character used to plot the kernel density curve if the histogram is produced on a line printer

You can request multiple kernel density estimates on the same histogram by specifying a list of values for either the C= or K= option. For more information, see the entries for these options. Also see Output 10.1.1 on page 273 and "Kernel Density Estimates" on page 319. By default, kernel density estimates are computed using the AMISE method.

L=*linetype*
specifies the line type used for fitted density curves. If used with the KERNEL option, you can specify a list of up to five line types for multiple kernel density estimates. See the entries for the C= and K= options for details on specifying multiple kernel density estimates. The default is 1, which produces a solid line.

LEGEND=*name* **| NONE**
Graphics
specifies the name of a LEGEND statement describing the legend for specification limit reference lines and fitted curves. Specifying LEGEND=NONE suppresses all legend information and is equivalent to specifying the NOLEGEND option.

LHREF=*linetype*
LH=*linetype*
Graphics
specifies the line type for lines requested with the HREF= option. See Output 11.1.1 on page 335. The default is 2, which produces a dashed line.

LOGNORMAL<**(***lognormal-options***)**>
displays a fitted lognormal density curve on the histogram. The curve equation is

$$p(x) = \begin{cases} \frac{h \times 100\%}{\sigma\sqrt{2\pi}(x-\theta)} \exp\left(-\frac{(\log(x-\theta)-\zeta)^2}{2\sigma^2}\right) & \text{for } x > \theta \\ 0 & \text{for } x \leq \theta \end{cases}$$

where

θ = threshold parameter
ζ = scale parameter
σ = shape parameter ($\sigma > 0$)
h = width of histogram interval

Note that the lognormal distribution is also referred to as the S_L distribution in the Johnson system of distributions.

The parameter θ for the lognormal distribution must be less than the minimum data value. You can specify θ with the THETA= *lognormal-option*. The default value for θ is zero. If you specify THETA=EST, a maximum likelihood estimate is computed for θ. You can specify the parameters σ and ζ with the SIGMA= and ZETA= *lognormal-options*. By default, maximum likelihood estimates are computed for σ and ζ. For example, the following statements fit a lognormal distribution function with a default value of $\theta = 0$ and with maximum likelihood estimates for σ and ζ:

```
proc capability;
   histogram length / lognormal;
run;
```

The LOGNORMAL option can appear only once in a HISTOGRAM statement. Table 11.2 on page 287 and Table 11.6 on page 288 list options that you can specify with the LOGNORMAL option. See Example 11.2 on page 336 and "Formulas for Fitted Curves" on page 313.

LOWER=*value-list*

specifies lower bounds for kernel density estimates requested with the KERNEL option. Enclose the LOWER= option in parentheses after the KERNEL option. You can specify up to five lower bounds for multiple kernel density estimates. If you specify more kernel estimates than lower bounds, the last lower bound is repeated for the remaining estimates.

LVREF=*linetype*
LV=*linetype*

specifies the line type for lines requested with the VREF= option. The default is 2, which produces a dashed line. *Graphics*

MAXITER=*n*

specifies the maximum number of iterations in the Newton-Raphson approximation of the maximum likelihood estimate of α for fitted gamma curves requested with the GAMMA option and c for fitted Weibull curves requested with the WEIBULL option. Enclose the MAXITER= option in parentheses after the GAMMA or WEIBULL option. The default is 20.

MIDPERCENTS

requests a table listing the midpoints and percent of observations in each histogram interval. For example, the following statements create the table in Figure 11.7:

```
proc capability;
   histogram length / midpercents;
run;
```

Histogram Bins for length

Bin Midpoint	Observed Percent
10.02	12.000
10.08	32.000
10.14	28.000
10.20	18.000
10.26	6.000
10.32	4.000

Figure 11.7. Table of Midpoints and Observed Percentages

If you specify the MIDPERCENTS option in parentheses after a density estimate option, a table listing the midpoints, observed percent of observations, and the estimated percent of the population in each interval (estimated from the fitted distribution) is printed.

The following statements create the table shown in Figure 11.8:

```
proc capability;
   histogram length / gamma(theta=3 midpercents);
run;
```

The CAPABILITY Procedure
Fitted Gamma Distribution for length

Histogram Bin Percents for Gamma Distribution

Bin Midpoint	Percent Observed	Percent Estimated
10.02	12.000	11.480
10.08	32.000	26.182
10.14	28.000	31.354
10.20	18.000	19.916
10.26	6.000	6.766
10.32	4.000	1.238

Figure 11.8. Table of Observed and Expected Percentages

MIDPOINTS=*value-list*

lists midpoints for the histogram intervals. The midpoints must be listed in increasing order and must be evenly spaced. The difference between consecutive midpoints is used as the width of the histogram bars. The same *value-list* is used for all variables. See Output 11.2.1 on page 338.

If you specify the MIDPOINTS= option, the range of the midpoints, extended at each end by half of the bar width, must cover the range of the data as well as any specification limits. For example, if you specify

```
midpoints=2 to 10 by 0.5
```

then all of the observations and specification limits must fall between 1.75 and 10.25 (otherwise, a default list of midpoints is used).

By default, the number of midpoints is determined using the algorithm described in Terrell and Scott (1985). The default midpoints are primarily applicable to continuous data that are approximately normally distributed.

If you display the histogram with a graphic device and use the MIDPOINTS= and HAXIS= options, you can use the ORDER= option in the AXIS statement you specified with the HAXIS= option. However, for the tick mark labels to coincide with the histogram interval midpoints, the range of the ORDER= list must encompass the range of the MIDPOINTS= list, as illustrated in the following statements:

```
proc capability;
   histogram length / midpoints=20 to 80 by 10
                      haxis=axis1;
   axis1 length=6 in order=10 20 30 40 50 60 70 80 90;
run;
```

MIDPTAXIS=*name*

is an alias for the HAXIS= option described earlier in this section. *Graphics*

MU=*value*

specifies the parameter μ for normal density curves requested with the NORMAL option. Enclose the MU= option in parentheses after the NORMAL option. The default value is the sample mean.

NAME='*string***'**

specifies a name for the plot, up to eight characters, that appears in the PROC GREPLAY master menu. The default is 'CAPABILI'. *Graphics*

NENDPOINTS=n

specifies the number of histogram interval endpoints and causes the endpoints, rather than interval midpoints, to be aligned with horizontal axis tick marks.

NMIDPOINTS=n

specifies the number of histogram intervals.

NOBARS

suppresses drawing of histogram bars. This option is useful when you want to display fitted curves only.

NOCURVELEGEND
NOCURVEL

suppresses the portion of the legend for fitted curves. If you use the INSET statement to display information about the fitted curve on the histogram, you can use the NOCURVELEGEND option to prevent the information about the fitted curve from being repeated in a legend at the bottom of the histogram. See Output 12.1.1 on page 375.

NOFRAME
suppresses the frame around the subplot area.

NOLEGEND
suppresses legends for specification limits, fitted curves, distribution lines, and hidden observations. See Example 11.6 on page 345. Specifying the NOLEGEND option is equivalent to specifying LEGEND=NONE.

NOPLOT
suppresses the creation of a plot. Use the NOPLOT option when you want only to print summary statistics for a fitted density or create either an OUTFIT= or an OUTHISTOGRAM= data set. See Example 11.4 on page 343.

NOPRINT
suppresses printed output summarizing the fitted curve. Enclose the NOPRINT option in parentheses following the distribution option. See "Customizing a Histogram" on page 284 for an example.

NORMAL<**(***normal-options***)**>
displays a fitted normal density curve on the histogram. The curve equation is

$$p(x) = \frac{h \times 100\%}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \quad \text{for } -\infty < x < \infty$$

where

μ = mean
σ = standard deviation ($\sigma > 0$)
h = width of histogram interval

Note that the normal distribution is also referred to as the S_N distribution in the Johnson system of distributions.

You can specify values for μ and σ with the MU= and SIGMA= *normal-options*, as shown in the following statements:

```
proc capability;
   histogram length / normal(mu=14 sigma=0.05);
run;
```

By default, the sample mean and sample standard deviation are used for μ and σ. The NORMAL option can appear only once in a HISTOGRAM statement. Table 11.2 (page 287) and Table 11.7 (page 288) list options that you can specify with the NORMAL option. See Figure 11.4 on page 283 and "Formulas for Fitted Curves" on page 313.

NOSPECLEGEND
NOSPECL
suppresses the portion of the legend for specification limit reference lines. See Figure 11.5 on page 284.

OUTFIT=*SAS-data-set*
creates a SAS data set that contains parameter estimates for fitted curves and related goodness-of-fit information. See "Output Data Sets" on page 328.

OUTHISTOGRAM=*SAS-data-set*
OUTHIST=*SAS-data-set*
creates a SAS data set that contains information about histogram intervals. Specifically, the data set contains the midpoints of the histogram intervals, the observed percent of observations in each interval, and the estimated percent of observations in each interval (estimated from each of the specified fitted curves). See "Output Data Sets" on page 328.

PCTAXIS=*name*|*value-list*
is an alias for the VAXIS= option. *Graphics*

PERCENTS=*value-list*
PERCENT=*value-list*
specifies a list of percents for which quantiles calculated from the data and quantiles estimated from the fitted curve are tabulated. The percents must be between 0 and 100. Enclose the PERCENTS= option in parentheses after the curve option. The default percents are 1, 5, 10, 25, 50, 75, 90, 95, and 99.

For example, the following statements create the table shown in Figure 11.9:

```
proc capability;
   histogram length / lognormal (percents=1 3 5 95 97 99);
run;
```

The CAPABILITY Procedure
Fitted Lognormal Distribution for length

Quantiles for Lognormal Distribution

Percent	Quantile Observed	Quantile Estimated
1.0	10.0180	9.95696
3.0	10.0180	9.98937
5.0	10.0310	10.00658
95.0	10.2780	10.24963
97.0	10.2930	10.26729
99.0	10.3220	10.30071

Figure 11.9. Estimated and Observed Quantiles for the Lognormal Curve

PFILL=*pattern*
specifies a pattern used to fill the bars of the histograms (or the areas under a fitted curve if you also specify the FILL option). See the entries for the CFILL= and FILL options for additional details. Refer to *SAS/GRAPH Software: Reference* for a list of pattern values. By default, the bars and curve areas are not filled.

RTINCLUDE
includes the right endpoint of each histogram interval in that interval. By default, the left endpoint is included in the histogram interval.

SB<**(***S_B-options* **)**>
displays a fitted Johnson S_B density curve on the histogram. The curve equation is

$$p(x) = \begin{cases} \frac{\delta h \times 100\%}{\sigma\sqrt{2\pi}} \left[\left(\frac{x-\theta}{\sigma}\right)\left(1-\frac{x-\theta}{\sigma}\right)\right]^{-1} \times \\ \exp\left[-\frac{1}{2}\left(\gamma + \delta \log\left(\frac{x-\theta}{\theta+\sigma-x}\right)\right)^2\right] & \text{for } \theta < x < \theta+\sigma \\ 0 & \text{for } x \le \theta \text{ or } x \ge \theta+\sigma \end{cases}$$

where

θ = threshold parameter $(-\infty < \theta < \infty)$
σ = scale parameter $(\sigma > 0)$
δ = shape parameter $(\delta > 0)$
γ = shape parameter $(-\infty < \gamma < \infty)$
h = width of histogram interval

The S_B distribution is bounded below by the parameter θ and above by the value $\theta + \sigma$. The parameter θ must be less than the minimum data value. You can specify θ with the THETA= S_B*-option*, or you can request that θ be estimated with the THETA = EST S_B*-option*. The default value for θ is zero. The sum $\theta + \sigma$ must be greater than the maximum data value. The default value for σ is one. You can specify σ with the SIGMA= S_B*-option*, or you can request that σ be estimated with the SIGMA = EST S_B*-option*. You can specify δ with the DELTA= S_B*-option*, and you can specify γ with the GAMMA= S_B*-option*. Note that the S_B*-options* are given in parentheses after the SB option.

By default, the method of percentiles is used to estimate the parameters of the S_B distribution. Alternatively, you can request the method of moments or the method of maximum likelihood with the FITMETHOD = MOMENTS or FITMETHOD = MLE options, respectively. Consider the following example:

```
proc capability;
   histogram length / sb;
   histogram length / sb( theta=est sigma=est );
   histogram length / sb( theta=0.5 sigma=8.4
                          delta=0.8 gamma=-0.6 );
run;
```

The first HISTOGRAM statement fits an S_B distribution with default values of $\theta = 0$ and $\sigma = 1$ and with percentile-based estimates for δ and γ. The second HISTOGRAM statement estimates all four parameters with the method of percentiles. The third HISTOGRAM statement displays an S_B curve with specified values for all four parameters.

The SB option can appear only once in a HISTOGRAM statement. Table 11.2 (page 287) and Table 11.8 (page 288) list options you can specify with the SB option.

SCALE=*value*

is an alias for the SIGMA= option for curves requested by the BETA, EXPONENTIAL, GAMMA, SB, SU, and WEIBULL options and an alias for the ZETA= option for curves requested by the LOGNORMAL option. See Example 11.1 on page 333.

SHAPE=*value*

is an alias for the ALPHA= option for curves requested with the GAMMA option, an alias for the SIGMA= option for curves requested with the LOGNORMAL option, and an alias for the C= option for curves requested with the WEIBULL option.

SIGMA=*value***|EST**

specifies the parameter σ for curves requested with the BETA, EXPONENTIAL, GAMMA, LOGNORMAL, NORMAL, SB, SU, and WEIBULL options. Enclose the SIGMA= option in parentheses after the distribution option. The following table summarizes the use of the SIGMA= option:

Distribution Keyword	SIGMA= Specifies	Default Value	Alias
BETA	scale parameter σ	1	SCALE=
EXPONENTIAL	scale parameter σ	maximum likelihood estimate	SCALE=
GAMMA	scale parameter σ	maximum likelihood estimate	SCALE=
LOGNORMAL	shape parameter σ	maximum likelihood estimate	SHAPE=
NORMAL	scale parameter σ	standard deviation	
SB	scale parameter σ	1	SCALE=
SU	scale parameter σ	percentile-based estimate	
WEIBULL	scale parameter σ	maximum likelihood estimate	SCALE=

With the BETA distribution option, you can specify SIGMA=EST to request a maximum likelihood estimate for σ. For syntax examples, see the entries for the BETA and NORMAL options.

SPECLEGEND=*name* **| NONE**

specifies the name of a LEGEND statement describing the legend for specification limits and fitted curves. Specifying SPECLEGEND=NONE, which suppresses the portion of the legend for specification limit references lines, is equivalent to specifying the NOSPECLEGEND option.

SU<**(***S_U-options* **)**>

displays a fitted Johnson S_U density curve on the histogram. The curve equation is

$$p(x) = \begin{cases} \frac{\delta h \times 100\%}{\sigma\sqrt{2\pi}} \frac{1}{\sqrt{1+((x-\theta)/\sigma)^2}} \times \\ \exp\left[-\frac{1}{2}\left(\gamma + \delta \sinh^{-1}\left(\frac{x-\theta}{\sigma}\right)\right)^2\right] & \text{for } x > \theta \\ 0 & \text{for } x \leq \theta \end{cases}$$

where

θ = location parameter $(-\infty < \theta < \infty)$
σ = scale parameter $(\sigma > 0)$
δ = shape parameter $(\delta > 0)$
γ = shape parameter $(-\infty < \gamma < \infty)$
h = width of histogram interval

You can specify the parameters with the THETA=, SIGMA=, DELTA=, and GAMMA= S_U*-options*, which are enclosed in parentheses after the SU option. If you do not specify these parameters, they are estimated.

By default, the method of percentiles is used to estimate the parameters of the S_U distribution. Alternatively, you can request the method of moments or the method of maximum likelihood with the FITMETHOD = MOMENTS or FITMETHOD = MLE options, respectively. Consider the following example:

```
proc capability;
   histogram length / su;
   histogram length / su( theta=0.5 sigma=8.4
                          delta=0.8 gamma=-0.6 );
run;
```

The first HISTOGRAM statement estimates all four parameters with the method of percentiles. The second HISTOGRAM statement displays an S_U curve with specified values for all four parameters.

The SU option can appear only once in a HISTOGRAM statement. Table 11.2 (page 287) and Table 11.9 (page 288) list options you can specify with the SU option.

SYMBOL='*character***'**

Line Printer

specifies the *character* used to plot the density curve or kernel density curve if the histogram is produced on a line printer. Enclose the SYMBOL= option in parentheses after the distribution option or the KERNEL option. The default character is the first letter of the distribution keyword or '1' for the first kernel density estimate, '2' for the second kernel density estimate, and so on. If you use the SYMBOL= option with the KERNEL option, you can specify a list of up to five characters in parentheses for multiple kernel density estimates. If there are more estimates than characters, the last character specified is used for the remaining estimates.

THETA=*value***|EST**

specifies the lower threshold parameter θ for curves requested with the BETA, EXPONENTIAL, GAMMA, LOGNORMAL, SB, and WEIBULL options, and the location parameter θ for curves requested with the SU option. Enclose the THETA= option in parentheses after the curve option. See Example 11.1 on page 333. The default *value* is zero. If you specify THETA=EST, an estimate is computed for θ.

THRESHOLD=*value*

is an alias for the THETA= option. See the preceding entry for the THETA= option.

UPPER=*value-list*

specifies upper bounds for kernel density estimates requested with the KERNEL option. Enclose the UPPER= option in parentheses after the KERNEL option. You can specify up to five upper bounds for multiple kernel density estimates. If you specify more kernel estimates than upper bounds, the last upper bound is repeated for the remaining estimates.

VAXIS=*name*|*value-list*

specifies the name of an AXIS statement describing the vertical axis. Alternatively, you can specify a *value-list* for the vertical axis. The PCTAXIS= option is an alias for the VAXIS= option. See Example 11.1 (page 333). *Graphics*

VMINOR=*n*
VM=*n*

specifies the number of minor tick marks between each major tick mark on the vertical axis. Minor tick marks are not labeled. The default is zero. *Graphics*

VREF=*value-list*

draws reference lines perpendicular to the vertical axis at the values specified. Also see the CVREF=, LVREF=, and VREFCHAR= options.

VREFCHAR='*character***'**

specifies the character used to form the lines requested by the VREF= option for a line printer. The default is a hyphen (-). *Line Printer*

VREFLABELS='*label1***' . . . '***labeln***'**
VREFLABEL='*label1***' . . . '***labeln***'**
VREFLAB='*label1***' . . . '***labeln***'**

specifies labels for the lines requested by the VREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can have up to 16 characters.

VSCALE=COUNT | PERCENT | PROPORTION

specifies the scale of the vertical axis. The value COUNT scales the data in units of the number of observations per data unit. The value PERCENT scales the data in units of percent of observations per data unit. The value PROPORTION scales the data in units of proportion of observations per data unit. See Figure 11.5 on page 284 for an illustration of VSCALE=COUNT. The default is PERCENT.

W=*n*

specifies the width in pixels of the fitted curve or the kernel density estimate curve. Enclose the W= option in parentheses after the distribution option or the KERNEL option (with the KERNEL option, you can specify a list of up to five W= values). For example, the following statements display a normal curve with a width of 3: *Graphics*

```
proc capability;
   histogram length / normal(w=3);
run;
```

The default is 1.

WBARLINE=*n*

specifies the width of bar outlines. By default, $n = 1$.

WEIBULL<**(***Weibull-options***)**>
displays a fitted Weibull density curve on the histogram. The curve equation is

$$p(x) = \begin{cases} \frac{ch \times 100\%}{\sigma}\left(\frac{x-\theta}{\sigma}\right)^{c-1}\exp\left(-\left(\frac{x-\theta}{\sigma}\right)^{c}\right) & \text{for } x > \theta \\ 0 & \text{for } x \leq \theta \end{cases}$$

where

θ = threshold parameter
σ = scale parameter $(\sigma > 0)$
c = shape parameter $(c > 0)$
h = width of histogram interval

The parameter θ must be less than the minimum data value. You can specify θ with the THETA= *Weibull-option*. The default value for θ is zero. If you specify THETA=EST, a maximum likelihood estimate is computed for θ. You can specify σ and c with the SIGMA= and C= *Weibull-options*. By default, maximum likelihood estimates are computed for c and σ. For example, the following statements fit a Weibull distribution with $\theta = 15$ and with maximum likelihood estimates for σ and c:

```
proc capability;
   histogram length / weibull(theta=15);
run;
```

Note that the maximum likelihood estimate of c is calculated iteratively using the Newton-Raphson approximation. The CDELTA=, CINITIAL=, and MAXITER= *Weibull-options* control the approximation.

The WEIBULL option can appear only once in a HISTOGRAM statement. Table 11.2 (page 287) and Table 11.10 (page 289) list the options that you can specify with the WEIBULL option. See Example 11.2 on page 336 and "Formulas for Fitted Curves" on page 313.

ZETA=*value*
specifies a value for the scale parameter ζ for lognormal density curves requested with the LOGNORMAL option. Enclose the ZETA= option in parentheses after the LOGNORMAL option. By default, the procedure calculates a maximum likelihood estimate for ζ. You can specify the SCALE= option as an alias for the ZETA= option.

Details

This section provides details on the following topics:

- formulas for fitted distributions
- formulas for kernel density estimates
- printed output
- OUTFIT= and OUTHISTOGRAM= data sets
- graphical enhancements to histograms

Formulas for Fitted Curves

The following sections provide information on the families of parametric distributions that you can fit with the HISTOGRAM statement. Properties of these distributions are discussed by Johnson *et al.* (1994, 1995).

Beta Distribution

The fitted density function is

$$p(x) = \begin{cases} \frac{(x-\theta)^{\alpha-1}(\sigma+\theta-x)^{\beta-1}}{B(\alpha,\beta)\sigma^{(\alpha+\beta-1)}} h \times 100\% & \text{for } \theta < x < \theta+\sigma \\ 0 & \text{for } x \leq \theta \text{ or } x \geq \theta+\sigma \end{cases}$$

where $B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$ and

$\theta =$ lower threshold parameter (lower endpoint parameter)
$\sigma =$ scale parameter $(\sigma > 0)$
$\alpha =$ shape parameter $(\alpha > 0)$
$\beta =$ shape parameter $(\beta > 0)$
$h =$ width of histogram interval

Note: This notation is consistent with that of other distributions that you can fit with the HISTOGRAM statement. However, many texts, including Johnson *et al.* (1995), write the beta density function as

$$p(x) = \begin{cases} \frac{(x-a)^{p-1}(b-x)^{q-1}}{B(p,q)(b-a)^{p+q-1}} & \text{for } a < x < b \\ 0 & \text{for } x \leq a \text{ or } x \geq b \end{cases}$$

The two notations are related as follows:

$\sigma = b - a$
$\theta = a$
$\alpha = p$
$\beta = q$

The range of the beta distribution is bounded below by a threshold parameter $\theta = a$ and above by $\theta + \sigma = b$. If you specify a fitted beta curve using the BETA option,

θ must be less than the minimum data value, and $\theta + \sigma$ must be greater than the maximum data value. You can specify θ and σ with the THETA= and SIGMA= *beta-options* in parentheses after the keyword BETA. By default, $\sigma = 1$ and $\theta = 0$. If you specify THETA=EST and SIGMA=EST, maximum likelihood estimates are computed for θ and σ.

In addition, you can specify α and β with the ALPHA= and BETA= *beta-options*, respectively. By default, the procedure calculates maximum likelihood estimates for α and β. For example, to fit a beta density curve to a set of data bounded below by 32 and above by 212 with maximum likelihood estimates for α and β, use the following statement:

```
histogram length / beta(theta=32 sigma=180);
```

The beta distributions are also referred to as Pearson Type I or II distributions. These include the *power-function* distribution ($\beta = 1$), the *arc-sine* distribution ($\alpha = \beta = \frac{1}{2}$), and the *generalized arc-sine* distributions ($\alpha + \beta = 1$, $\beta \neq \frac{1}{2}$).

You can use the DATA step function BETAINV to compute beta quantiles and the DATA step function PROBBETA to compute beta probabilities.

Exponential Distribution

The fitted density function is

$$p(x) = \begin{cases} \frac{h \times 100\%}{\sigma} \exp(-(\frac{x-\theta}{\sigma})) & \text{for } x \geq \theta \\ 0 & \text{for } x < \theta \end{cases}$$

where

$\theta =$ threshold parameter
$\sigma =$ scale parameter ($\sigma > 0$)
$h =$ width of histogram interval

The threshold parameter θ must be less than or equal to the minimum data value. You can specify θ with the THRESHOLD= *exponential-option*. By default, $\theta = 0$. If you specify THETA=EST, a maximum likelihood estimate is computed for θ. In addition, you can specify σ with the SCALE= *exponential-option*. By default, the procedure calculates a maximum likelihood estimate for σ. Note that some authors define the scale parameter as $\frac{1}{\sigma}$.

The exponential distribution is a special case of both the gamma distribution (with $\alpha = 1$) and the Weibull distribution (with $c = 1$). A related distribution is the *extreme value* distribution. If $Y = \exp(-X)$ has an exponential distribution, then X has an extreme value distribution.

Gamma Distribution

The fitted density function is

$$p(x) = \begin{cases} \frac{h \times 100\%}{\Gamma(\alpha)\sigma}\left(\frac{x-\theta}{\sigma}\right)^{\alpha-1}\exp\left(-\left(\frac{x-\theta}{\sigma}\right)\right) & \text{for } x > \theta \\ 0 & \text{for } x \leq \theta \end{cases}$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)
α = shape parameter ($\alpha > 0$)
h = width of histogram interval

The threshold parameter θ must be less than the minimum data value. You can specify θ with the THRESHOLD= *gamma-option*. By default, $\theta = 0$. If you specify THETA=EST, a maximum likelihood estimate is computed for θ. In addition, you can specify σ and α with the SCALE= and ALPHA= *gamma-options*. By default, the procedure calculates maximum likelihood estimates for σ and α.

The gamma distributions are also referred to as Pearson Type III distributions, and they include the chi-square, exponential, and Erlang distributions. The probability density function for the chi-square distribution is

$$p(x) = \begin{cases} \frac{1}{2\Gamma\left(\frac{\nu}{2}\right)}\left(\frac{x}{2}\right)^{\frac{\nu}{2}-1}\exp\left(-\frac{x}{2}\right) & \text{for } x > 0 \\ 0 & \text{for } x \leq 0 \end{cases}$$

Notice that this is a gamma distribution with $\alpha = \frac{\nu}{2}$, $\sigma = 2$, and $\theta = 0$. The exponential distribution is a gamma distribution with $\alpha = 1$, and the Erlang distribution is a gamma distribution with α being a positive integer. A related distribution is the Rayleigh distribution. If $R = \frac{\max(X_1,\ldots,X_n)}{\min(X_1,\ldots,X_n)}$ where the X_i's are independent χ^2_ν variables, then $\log R$ is distributed with a χ_ν distribution having a probability density function of

$$p(x) = \begin{cases} \left[2^{\frac{\nu}{2}-1}\Gamma\left(\frac{\nu}{2}\right)\right]^{-1} x^{\nu-1}\exp\left(-\frac{x^2}{2}\right) & \text{for } x > 0 \\ 0 & \text{for } x \leq 0 \end{cases}$$

If $\nu = 2$, the preceding distribution is referred to as the Rayleigh distribution.

You can use the DATA step function GAMINV to compute gamma quantiles and the DATA step function PROBGAM to compute gamma probabilities.

Johnson S_B Distribution

The fitted density function is

$$p(x) = \begin{cases} \frac{\delta h \times 100\%}{\sigma\sqrt{2\pi}}\left[\left(\frac{x-\theta}{\sigma}\right)\left(1-\frac{x-\theta}{\sigma}\right)\right]^{-1} \times \\ \exp\left[-\frac{1}{2}\left(\gamma + \delta\log\left(\frac{x-\theta}{\theta+\sigma-x}\right)\right)^2\right] & \text{for } \theta < x < \theta+\sigma \\ 0 & \text{for } x \leq \theta \text{ or } x \geq \theta+\sigma \end{cases}$$

where

θ = threshold parameter $(-\infty < \theta < \infty)$
σ = scale parameter $(\sigma > 0)$
δ = shape parameter $(\delta > 0)$
γ = shape parameter $(-\infty < \gamma < \infty)$
h = width of histogram interval

The S_B distribution is bounded below by the parameter θ and above by the value $\theta + \sigma$. The parameter θ must be less than the minimum data value. You can specify θ with the THETA= S_B-*option*, or you can request that θ be estimated with the THETA = EST S_B-*option*. The default value for θ is zero. The sum $\theta + \sigma$ must be greater than the maximum data value. The default value for σ is one. You can specify σ with the SIGMA= S_B-*option*, or you can request that σ be estimated with the SIGMA = EST S_B-*option*.

By default, the method of percentiles given by Slifker and Shapiro (1980) is used to estimate the parameters. This method is based on four data percentiles, denoted by x_{-3z}, x_{-z}, x_z, and x_{3z}, which correspond to the four equally spaced percentiles of a standard normal distribution, denoted by $-3z$, $-z$, z, and $3z$, under the transformation

$$z = \gamma + \delta \log\left(\frac{x - \theta}{\theta + \sigma - x}\right)$$

The default value of z is 0.524. The results of the fit are dependent on the choice of z, and you can specify other values with the FITINTERVAL= option (specified in parentheses after the SB option). If you use the method of percentiles, you should select a value of z that corresponds to percentiles which are critical to your application.

The following values are computed from the data percentiles:

$$\begin{aligned} m &= x_{3z} - x_z \\ n &= x_{-z} - x_{-3z} \\ p &= x_z - x_{-z} \end{aligned}$$

It was demonstrated by Slifker and Shapiro (1980) that

$\frac{mn}{p^2} > 1$ for any S_U distribution
$\frac{mn}{p^2} < 1$ for any S_B distribution
$\frac{mn}{p^2} = 1$ for any S_L (lognormal) distribution

A tolerance interval around one is used to discriminate among the three families with this ratio criterion. You can specify the tolerance with the FITTOLERANCE= option (specified in parentheses after the SB option). The default tolerance is 0.01. Assuming that the criterion satisfies the inequality

$$\frac{mn}{p^2} < 1 - \text{tolerance}$$

the parameters of the S_B distribution are computed using the explicit formulas derived by Slifker and Shapiro (1980).

If you specify FITMETHOD = MOMENTS (in parentheses after the SB option) the method of moments is used to estimate the parameters. If you specify FITMETHOD = MLE (in parentheses after the SB option) the method of maximum likelihood is used to estimate the parameters. Note that maximum likelihood estimates may not always exist. Refer to Bowman and Shenton (1983) for discussion of methods for fitting Johnson distributions.

Johnson S_U Distribution

The fitted density function is

$$p(x) = \begin{cases} \frac{\delta h \times 100\%}{\sigma\sqrt{2\pi}} \frac{1}{\sqrt{1+((x-\theta)/\sigma)^2}} \times \\ \exp\left[-\frac{1}{2}\left(\gamma + \delta \sinh^{-1}\left(\frac{x-\theta}{\sigma}\right)\right)^2\right] & \text{for } x > \theta \\ 0 & \text{for } x \leq \theta \end{cases}$$

where

$\theta =$ location parameter $(-\infty < \theta < \infty)$
$\sigma =$ scale parameter $(\sigma > 0)$
$\delta =$ shape parameter $(\delta > 0)$
$\gamma =$ shape parameter $(-\infty < \gamma < \infty)$
$h =$ width of histogram interval

You can specify the parameters with the THETA=, SIGMA=, DELTA=, and GAMMA= S_U*-options*, which are enclosed in parentheses after the SU option. If you do not specify these parameters, they are estimated.

By default, the method of percentiles given by Slifker and Shapiro (1980) is used to estimate the parameters. This method is based on four data percentiles, denoted by x_{-3z}, x_{-z}, x_z, and x_{3z}, which correspond to the four equally spaced percentiles of a standard normal distribution, denoted by $-3z$, $-z$, z, and $3z$, under the transformation

$$z = \gamma + \delta \sinh^{-1}\left(\frac{x-\theta}{\sigma}\right)$$

The default value of z is 0.524. The results of the fit are dependent on the choice of z, and you can specify other values with the FITINTERVAL= option (specified in parentheses after the SB option). If you use the method of percentiles, you should select a value of z that corresponds to percentiles which are critical to your application. You can specify the value of z with the FITINTERVAL= option (specified in parentheses after the SU option).

The following values are computed from the data percentiles:

$$\begin{aligned} m &= x_{3z} - x_z \\ n &= x_{-z} - x_{-3z} \\ p &= x_z - x_{-z} \end{aligned}$$

It was demonstrated by Slifker and Shapiro (1980) that

$$\begin{array}{ll} \frac{mn}{p^2} > 1 & \text{for any } S_U \text{ distribution} \\ \frac{mn}{p^2} < 1 & \text{for any } S_B \text{ distribution} \\ \frac{mn}{p^2} = 1 & \text{for any } S_L \text{ (lognormal) distribution} \end{array}$$

A tolerance interval around one is used to discriminate among the three families with this ratio criterion. You can specify the tolerance with the FITTOLERANCE= option (specified in parentheses after the SU option). The default tolerance is 0.01. Assuming that the criterion satisfies the inequality

$$\frac{mn}{p^2} > 1 + \text{tolerance}$$

the parameters of the S_U distribution are computed using the explicit formulas derived by Slifker and Shapiro (1980).

If you specify FITMETHOD = MOMENTS (in parentheses after the SU option) the method of moments is used to estimate the parameters. If you specify FITMETHOD = MLE (in parentheses after the SU option) the method of maximum likelihood is used to estimate the parameters. Note that maximum likelihood estimates may not always exist. Refer to Bowman and Shenton (1983) for discussion of methods for fitting Johnson distributions.

Lognormal Distribution

The fitted density function is

$$p(x) = \begin{cases} \frac{h \times 100\%}{\sigma\sqrt{2\pi}(x-\theta)} \exp\left(-\frac{(\log(x-\theta)-\zeta)^2}{2\sigma^2}\right) & \text{for } x > \theta \\ 0 & \text{for } x \leq \theta \end{cases}$$

where

$\theta =$ threshold parameter
$\zeta =$ scale parameter $(-\infty < \zeta < \infty)$
$\sigma =$ shape parameter $(\sigma > 0)$
$h =$ width of histogram interval The threshold parameter θ must be less than the minimum data value. You can specify θ with the THRESHOLD= *lognormal-option*. By default, $\theta = 0$. If you specify THETA=EST, a maximum likelihood estimate is computed for θ. You can specify ζ and σ with the SCALE= and SHAPE= *lognormal-options*, respectively. By default, the procedure calculates maximum likelihood estimates for these parameters.

Note: The lognormal distribution is also referred to as the S_L distribution in the Johnson system of distributions.

Note: This book uses σ to denote the shape parameter of the lognormal distribution, whereas σ is used to denote the scale parameter of the beta, exponential, gamma, normal, and Weibull distributions. The use of σ to denote the lognormal shape parameter is based on the fact that $\frac{1}{\sigma}(\log(X-\theta)-\zeta)$ has a standard normal distribution if X is lognormally distributed.

Normal Distribution

The fitted density function is

$$p(x) = \frac{h \times 100\%}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \quad \text{for } -\infty < x < \infty$$

where

μ = mean
σ = standard deviation ($\sigma > 0$)
h = width of histogram interval

You can specify μ and σ with the MU= and SIGMA= *normal-options*, respectively. By default, the procedure estimates μ with the sample mean and σ with the sample standard deviation.

You can use the DATA step function PROBIT to compute normal quantiles and the DATA step function PROBNORM to compute probabilities.

Note: The normal distribution is also referred to as the S_N distribution in the Johnson system of distributions.

Weibull Distribution

The fitted density function is

$$p(x) = \begin{cases} \frac{ch \times 100\%}{\sigma}\left(\frac{x-\theta}{\sigma}\right)^{c-1} \exp\left(-\left(\frac{x-\theta}{\sigma}\right)^c\right) & \text{for } x > \theta \\ 0 & \text{for } x \leq \theta \end{cases}$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)
c = shape parameter ($c > 0$)
h = width of histogram interval

The threshold parameter θ must be less than the minimum data value. You can specify θ with the THRESHOLD= *Weibull-option*. By default, $\theta = 0$. If you specify THETA=EST, a maximum likelihood estimate is computed for θ. You can specify σ and c with the SCALE= and SHAPE= *Weibull-options*, respectively. By default, the procedure calculates maximum likelihood estimates for σ and c.

The exponential distribution is a special case of the Weibull distribution where $c = 1$.

Kernel Density Estimates

You can use the KERNEL option to superimpose kernel density estimates on histograms. Smoothing the data distribution with a kernel density estimate can be more effective than using a histogram to examine features that might be obscured by the choice of histogram bins or sampling variation. A kernel density estimate can also be more effective than a parametric curve fit when the process distribution is multimodal. See Example 11.5 on page 344.

The general form of the kernel density estimator is

$$\hat{f}_\lambda(x) = \frac{1}{n\lambda}\sum_{i=1}^{n} K_0\left(\frac{x - x_i}{\lambda}\right)$$

where $K_0(\cdot)$ is a kernel function, λ is the bandwidth, n is the sample size, and x_i is the i^{th} observation.

The KERNEL option provides three kernel functions (K_0): normal, quadratic, and triangular. You can specify the function with the K= *kernel-option* in parentheses after the KERNEL option. Values for the K= option are NORMAL, QUADRATIC, and TRIANGULAR (with aliases of N, Q, and T, respectively). By default, a normal kernel is used. The formulas for the kernel functions are

Normal	$K_0(t) = \frac{1}{\sqrt{2\pi}}\exp(-\frac{1}{2}t^2)$	for $-\infty < t < \infty$
Quadratic	$K_0(t) = \frac{3}{4}(1 - t^2)$	for $\lvert t\rvert \le 1$
Triangular	$K_0(t) = 1 - \lvert t\rvert$	for $\lvert t\rvert \le 1$

The value of λ, referred to as the bandwidth parameter, determines the degree of smoothness in the estimated density function. You specify λ indirectly by specifying a standardized bandwidth c with the C= *kernel-option*. If Q is the interquartile range, and n is the sample size, then c is related to λ by the formula

$$\lambda = cQn^{-\frac{1}{5}}$$

For a specific kernel function, the discrepancy between the density estimator $\hat{f}_\lambda(x)$ and the true density $f(x)$ is measured by the mean integrated square error (MISE):

$$\text{MISE}(\lambda) = \int_x \{E(\hat{f}_\lambda(x)) - f(x)\}^2 dx + \int_x var(\hat{f}_\lambda(x))dx$$

The MISE is the sum of the integrated squared bias and the variance. An approximate mean integrated square error (AMISE) is

$$\text{AMISE}(\lambda) = \frac{1}{4}\lambda^4\left(\int_t t^2K(t)dt\right)^2\int_x \left(f''(x)\right)^2 dx + \frac{1}{n\lambda}\int_t K(t)^2 dt$$

A bandwidth that minimizes AMISE can be derived by treating $f(x)$ as the normal density having parameters μ and σ estimated by the sample mean and standard deviation. If you do not specify a bandwidth parameter or if you specify C=MISE, the bandwidth that minimizes AMISE is used. The value of AMISE can be used to compare different density estimates. For each estimate, the bandwidth parameter c, the kernel function type, and the value of AMISE are reported in the SAS log.

The general kernel density estimates assume that the domain of the density to estimate can take on all values on a real line. However, sometimes the domain of a density is an

interval bounded on one or both sides. For example, if a variable Y is a measurement of only positive values, then the kernel density curve should be bounded so that it is zero for negative Y values.

The CAPABILITY procedure uses a reflection technique to create the bounded kernel density curve, as described in Silverman (1986, pp. 30-31). It adds the reflections of kernel density that are outside the boundary to the bounded kernel estimates. The general form of the bounded kernel density estimator is computed by replacing $K_0\left(\frac{x-x_i}{\lambda}\right)$ in the original equation with

$$\left\{K_0\left(\frac{x-x_i}{\lambda}\right)+K_0\left(\frac{(x-x_l)+(x_i-x_l)}{\lambda}\right)+K_0\left(\frac{(x_u-x)+(x_u-x_i)}{\lambda}\right)\right\}$$

where x_l is the lower bound and x_u is the upper bound.

Without a lower bound, $x_l = \infty$ and $K_0\left(\frac{(x-x_l)+(x_i-x_l)}{\lambda}\right)$ equals zero. Similarly, without an upper bound, $x_u = \infty$ and $K_0\left(\frac{(x_u-x)+(x_u-x_i)}{\lambda}\right)$ equals zero.

When C=MISE is used with a bounded kernel density, the CAPABILITY procedure uses a bandwidth that minimizes the AMISE for its corresponding unbounded kernel.

Printed Output

If you request a fitted parametric distribution, printed output summarizing the fit is produced in addition to the graphical display. Figure 11.10 shows the printed output for a fitted lognormal distribution requested by the following statements:

```
proc capability data=hang;
   spec target=14 lsl=13.95 usl=14.05;
   hist / lognormal(color=black indices midpercents);
run;
```

The summary is organized into the following parts:

- Parameters
- Chi-Square Goodness-of-Fit Test
- EDF Goodness-of-Fit Tests
- Specifications
- Indices Using the Fitted Curve
- Histogram Intervals
- Quantiles

These parts are described in the sections that follow.

Parameters

This section lists the parameters for the fitted curve as well as the estimated mean and estimated standard deviation. See "Formulas for Fitted Curves" on page 313.

The CAPABILITY Procedure
Fitted Lognormal Distribution for width

Parameters for Lognormal Distribution

Parameter	Symbol	Estimate
Threshold	Theta	0
Scale	Zeta	2.638966
Shape	Sigma	0.001497
Mean		13.99873
Std Dev		0.020952

Goodness-of-Fit Tests for Lognormal Distribution

Test	----Statistic-----		DF	------p Value------	
Kolmogorov-Smirnov	D	0.09148348		Pr > D	>0.150
Cramer-von Mises	W-Sq	0.05040427		Pr > W-Sq	>0.500
Anderson-Darling	A-Sq	0.33476355		Pr > A-Sq	>0.500
Chi-Square	Chi-Sq	2.87938822	3	Pr > Chi-Sq	0.411

Capability Indices Based on Lognormal Distribution

Cp	0.795463
CPL	0.776822
CPU	0.814021
Cpk	0.776822
Cpm	0.792237

Histogram Bin Percents for Lognormal Distribution

Bin	-------Percent------	
Midpoint	Observed	Estimated
13.95	4.000	2.963
13.97	18.000	15.354
13.99	26.000	33.872
14.01	38.000	32.055
14.03	10.000	13.050
14.05	4.000	2.281

Quantiles for Lognormal Distribution

	------Quantile------	
Percent	Observed	Estimated
1.0	13.9440	13.9501
5.0	13.9656	13.9643
10.0	13.9710	13.9719
25.0	13.9860	13.9846
50.0	14.0018	13.9987
75.0	14.0129	14.0129
90.0	14.0218	14.0256
95.0	14.0241	14.0332
99.0	14.0470	14.0475

Figure 11.10. Sample Summary of Fitted Distribution

Chi-Square Goodness-of-Fit Test

The chi-square goodness-of-fit statistic for a fitted parametric distribution is computed as follows:

$$\chi^2 = \sum_{i=1}^{m} \frac{(O_i - E_i)^2}{E_i}$$

where

O_i = observed value in i^{th} histogram interval
E_i = expected value in i^{th} histogram interval
m = number of histogram intervals
p = number of estimated parameters

The degrees of freedom for the chi-square test is equal to $m-p-1$. You can save the observed and expected interval values in the OUTFIT= data set discussed in "Output Data Sets" on page 328.

Note that empty intervals are not combined, and the range of intervals used to compute χ^2 begins with the first interval containing observations and ends with the final interval containing observations.

EDF Goodness-of-Fit Tests

When you fit a parametric distribution, the HISTOGRAM statement provides a series of goodness-of-fit tests based on the empirical distribution function (EDF). The EDF tests offer advantages over the chi-square goodness-of-fit test, including improved power and invariance with respect to the histogram midpoints. For a thorough discussion, refer to D'Agostino and Stephens (1986).

The empirical distribution function is defined for a set of n independent observations $X_1, \ldots, X_n$ with a common distribution function $F(x)$. Denote the observations ordered from smallest to largest as $X_{(1)}, \ldots, X_{(n)}$. The empirical distribution function, $F_n(x)$, is defined as

$$\begin{aligned} F_n(x) &= 0, \quad x < X_{(1)} \\ F_n(x) &= \tfrac{i}{n}, \quad X_{(i)} \le x < X_{(i+1)} \quad i = 1, \ldots, n-1 \\ F_n(x) &= 1, \quad X_{(n)} \le x \end{aligned}$$

Note that $F_n(x)$ is a step function that takes a step of height $\frac{1}{n}$ at each observation. This function estimates the distribution function $F(x)$. At any value x, $F_n(x)$ is the proportion of observations less than or equal to x, while $F(x)$ is the probability of an observation less than or equal to x. EDF statistics measure the discrepancy between $F_n(x)$ and $F(x)$.

The computational formulas for the EDF statistics make use of the probability integral transformation $U = F(X)$. If $F(X)$ is the distribution function of X, the random variable U is uniformly distributed between 0 and 1.

Given n observations $X_{(1)}, \ldots, X_{(n)}$, the values $U_{(i)} = F(X_{(i)})$ are computed by applying the transformation, as shown in the following sections.

The HISTOGRAM statement provides three EDF tests:

- Kolmogorov-Smirnov
- Anderson-Darling
- Cramér-von Mises

These tests are based on various measures of the discrepancy between the empirical distribution function $F_n(x)$ and the proposed parametric cumulative distribution function $F(x)$.

The following sections provide formal definitions of the EDF statistics.

Kolmogorov-Smirnov Statistic

The Kolmogorov-Smirnov statistic (D) is defined as

$$D = \sup_x |F_n(x) - F(x)|$$

The Kolmogorov-Smirnov statistic belongs to the supremum class of EDF statistics. This class of statistics is based on the largest vertical difference between $F(x)$ and $F_n(x)$.

The Kolmogorov-Smirnov statistic is computed as the maximum of D^+ and D^-, where D^+ is the largest vertical distance between the EDF and the distribution function when the EDF is greater than the distribution function, and D^- is the largest vertical distance when the EDF is less than the distribution function.

$$\begin{aligned} D^+ &= \max_i \left(\frac{i}{n} - U_{(i)}\right) \\ D^- &= \max_i \left(U_{(i)} - \frac{i-1}{n}\right) \\ D &= \max\left(D^+, D^-\right) \end{aligned}$$

Anderson-Darling Statistic

The Anderson-Darling statistic and the Cramér-von Mises statistic belong to the quadratic class of EDF statistics. This class of statistics is based on the squared difference $(F_n(x) - F(x))^2$. Quadratic statistics have the following general form:

$$Q = n \int_{-\infty}^{+\infty} (F_n(x) - F(x))^2 \, \psi(x) dF(x)$$

The function $\psi(x)$ weights the squared difference $(F_n(x) - F(x))^2$.

The Anderson-Darling statistic (A^2) is defined as

$$A^2 = n \int_{-\infty}^{+\infty} (F_n(x) - F(x))^2 \left[F(x)\left(1 - F(x)\right)\right]^{-1} dF(x)$$

Here the weight function is $\psi(x) = [F(x)\,(1 - F(x))]^{-1}$.

The Anderson-Darling statistic is computed as

$$A^2 = -n - \frac{1}{n}\sum_{i=1}^{n}\left[(2i-1)\log U_{(i)} + (2n+1-2i)\log\left(\{1 - U_{(i)}\right)\right]$$

Cramér-von Mises Statistic

The Cramér-von Mises statistic (W^2) is defined as

$$W^2 = n\int_{-\infty}^{+\infty}(F_n(x) - F(x))^2\,dF(x)$$

Here the weight function is $\psi(x) = 1$.

The Cramér-von Mises statistic is computed as

$$W^2 = \sum_{i=1}^{n}\left(U_{(i)} - \frac{2i-1}{2n}\right)^2 + \frac{1}{12n}$$

Probability Values for EDF Tests

Once the EDF test statistics are computed, the associated probability values (p-values) must be calculated. The CAPABILITY procedure uses internal tables of probability levels similar to those given by D'Agostino and Stephens (1986). If the value is between two probability levels, then linear interpolation is used to estimate the probability value.

The probability value depends upon the parameters that are known and the parameters that are estimated for the distribution you are fitting. Table 11.17 summarizes different combinations of estimated parameters for which EDF tests are available.

Specifications

This section is included in the summary only if you provide specification limits, and it tabulates the limits as well as the observed percentages and estimated percentages outside the limits.

The estimated percentages are computed only if fitted distributions are requested and are based on the probability that an observed value exceeds the specification limits, assuming the fitted distribution. The observed percentages are the percents of observations outside the specification limits.

Indices Using Fitted Curves

This section is included in the summary only if you specify the INDICES option in parentheses after a distribution option, as in the statements on page 321 that produce Figure 11.10. Standard process capability indices, such as C_p and C_{pk}, are not appropriate if the data are not normally distributed. The INDICES option computes generalizations of the standard indices using the fact that for the normal distribution, 3σ

Table 11.17. Availability of EDF Tests

Distribution	Parameters			Tests Available
	Threshold	Scale	Shape	
Beta	θ known	σ known	α, β known	all
	θ known	σ known	$\alpha, \beta < 5$ unknown	all
Exponential	θ known,	σ known		all
	θ known	σ unknown		all
	θ unknown	σ known		all
	θ unknown	σ unknown		all
Gamma	θ known	σ known	α known	all
	θ known	σ unknown	α known	all
	θ known	σ known	α unknown	all
	θ known	σ unknown	α unknown	all
	θ unknown	σ known	$\alpha > 1$ known	all
	θ unknown	σ unknown	$\alpha > 1$ known	all
	θ unknown	σ known	$\alpha > 1$ unknown	all
	θ unknown	σ unknown	$\alpha > 1$ unknown	all
Lognormal	θ known	ζ known	σ known	all
	θ known	ζ known	σ unknown	A^2 and W^2
	θ known	ζ unknown	σ known	A^2 and W^2
	θ known	ζ unknown	σ unknown	all
	θ unknown	ζ known	$\sigma < 3$ known	all
	θ unknown	ζ known	$\sigma < 3$ unknown	all
	θ unknown	ζ unknown	$\sigma < 3$ known	all
	θ unknown	ζ unknown	$\sigma < 3$ unknown	all
Normal	θ known	σ known		all
	θ known	σ unknown		A^2 and W^2
	θ unknown	σ known		A^2 and W^2
	θ unknown	σ unknown		all
Weibull	θ known	σ known	c known	all
	θ known	σ unknown	c known	A^2 and W^2
	θ known	σ known	c unknown	A^2 and W^2
	θ known	σ unknown	c unknown	A^2 and W^2
	θ unknown	σ known	$c > 2$ known	all
	θ unknown	σ unknown	$c > 2$ known	all
	θ unknown	σ known	$c > 2$ unknown	all
	θ unknown	σ unknown	$c > 2$ unknown	all

is both the distance from the lower 0.135 percentile to the median (or mean) and the distance from the median (or mean) to the upper 99.865 percentile. These percentiles are estimated from the fitted distribution, and the appropriate percentile-to-median distances are substituted for 3σ in the standard formulas.

Writing T for the target, *LSL* and *USL* for the lower and upper specification limits, and P_α for the $100\alpha^{\text{th}}$ percentile, the generalized capability indices are as follows:

$$C_{pl} = \frac{P_{0.5} - LSL}{P_{0.5} - P_{0.00135}}$$

$$C_{pu} = \frac{USL - P_{0.5}}{P_{0.99865} - P_{0.5}}$$

$$C_p = \frac{USL - LSL}{P_{0.99865} - P_{0.00135}}$$

$$C_{pk} = \min\left(\frac{P_{0.5} - \text{LSL}}{P_{0.5} - P_{0.00135}}, \frac{\text{USL} - P_{0.5}}{P_{0.99865} - P_{0.5}}\right)$$

$$K = 2 \times \frac{\left|\frac{1}{2}(\text{USL} + \text{LSL}) - P_{0.5}\right|}{\text{USL} - \text{LSL}}$$

$$C_{pm} = \frac{\min\left(\frac{T - LSL}{P_{0.5} - P_{0.00135}}, \frac{USL - T}{P_{0.99865} - P_{0.5}}\right)}{\sqrt{1 + \left(\frac{\mu - T}{\sigma}\right)^2}}$$

If the data are normally distributed, these formulas reduce to the formulas for the standard capability indices, which are given on page 204.

The following guidelines apply to the use of generalized capability indices requested with the INDICES option:

- When you choose the family of parametric distributions for the fitted curve, consider whether an appropriate family can be derived from assumptions about the process.
- Whenever possible, examine the data distribution with a histogram, probability plot, or quantile-quantile plot.
- Apply goodness-of-fit tests to assess how well the parametric distribution models the data.
- Consider whether a generalized index has a meaningful practical interpretation in your application.

At the time of this writing, there is ongoing research concerning the application of generalized capability indices, and it is important to note that other approaches can be used with nonnormal data:

- Transform the data to normality, then compute and report standard capability indices on the transformed scale.
- Report the proportion of nonconforming output estimated from the fitted distribution.
- If it is not possible to adequately model the data distribution with a parametric density, smooth the data distribution with a kernel density estimate and simply report the proportion of nonconforming output.

Refer to Rodriguez (1992) for additional discussion.

Histogram Intervals

This section is included in the summary only if you specify the MIDPERCENTS option in parentheses after the distribution option, as in the statements on page 321 that produce Figure 11.10. This table lists the interval midpoints along with the observed and estimated percentages of the observations that lie in the interval. The estimated percentages are based on the fitted distribution.

In addition, you can specify the MIDPERCENTS option to request a table of interval midpoints with the observed percent of observations that lie in the interval. See the entry for the MIDPERCENTS option on page 303.

Quantiles

This table lists observed and estimated quantiles. You can use the PERCENTS= option to specify the list of quantiles to appear in this list. The list in Figure 11.10 is the default list. See the entry for the PERCENTS= option on page 307.

Output Data Sets

You can create two output data sets with the HISTOGRAM statement: the OUTFIT= data set and the OUTHISTOGRAM= data set. These data sets are described in the following sections.

OUTFIT= Data Sets

The OUTFIT= data set contains the parameters of fitted density curves, information on chi-square and EDF goodness-of-fit tests, specification limit information, and capability indices based on the fitted distribution. Since you can specify multiple HISTOGRAM statements with the CAPABILITY procedure, you can create several OUTFIT= data sets. For each variable plotted with the HISTOGRAM statement, the OUTFIT= data set contains one observation for each fitted distribution requested in the HISTOGRAM statement. If you use a BY statement, the OUTFIT= data set contains several observations for each BY group (one observation for each variable and fitted density combination). ID variables are not saved in the OUTFIT= data set.

The OUTFIT= data set contains the variables listed in Table 11.18 on page 329. By default, an OUTFIT= data set contains _MIDPT1_ and _MIDPTN_ variables, whose values identify histogram intervals by their midpoints. When the ENDPOINTS= or NENDPOINTS option is specified, intervals are identified by endpoint values instead.

If the RTINCLUDE option is specified, the variables _MAXPT1_ and _MAXPTN_ contain upper endpoint values. Otherwise, the variables _MINPT1_ and _MINPTN_ contain lower endpoint values.

Table 11.18. Variables in the OUTFIT= Data Set

Variable	Description
ADASQ	Anderson-Darling EDF goodness-of-fit statistic
ADP	p-value for Anderson-Darling EDF goodness-of-fit test
CHISQ	chi-square goodness-of-fit statistic
_CP_	generalized capability index C_p based on the fitted curve
CPK	generalized capability index C_{pk} based on the fitted curve
CPL	generalized capability index *CPL* based on the fitted curve
CPM	generalized capability index C_{pm} based on the fitted curve
CPU	generalized capability index *CPU* based on the fitted curve
CURVE	name of fitted distribution (abbreviated to 8 characters)
CVMWSQ	Cramer-von Mises EDF goodness-of-fit statistic
CVMP	p-value for Cramer-von Mises EDF goodness-of-fit test
_DF_	degrees of freedom for chi-square goodness-of-fit test
ESTGTR	estimated percent of population greater than upper specification limit
ESTLSS	estimated percent of population less than lower specification limit
ESTSTD	estimated standard deviation
EXPECT	estimated mean
_K_	generalized capability index K based on the fitted curve
KSD	Kolmogorov-Smirnov EDF goodness-of-fit statistic
KSP	p-value for Kolmogorov-Smirnov EDF goodness-of-fit test
LOCATN	location parameter for fitted distribution. For the normal distribution, this is either the value of μ specified with the MU= option or the sample mean. For all other distributions, this is either the value specified with the THRESHOLD= option or zero.
LSL	lower specification limit
MAXPT1	upper endpoint of first interval used to calculate the value of the chi-square statistic.
MAXPTN	upper endpoint of last interval used to calculate the value of the chi-square statistic.
MIDPT1	midpoint of first interval used to calculate the value of the chi-square statistic. This is the leftmost interval that contains at least one value of the variable.
MIDPTN	midpoint of last interval used to calculate the value of the chi-square statistic. This is the rightmost interval that contains at least one value of the variable.

Table 11.18. *(continued)*

Variable	Description
MINPT1	lower endpoint of first interval used to calculate the value of the chi-square statistic.
MINPTN	lower endpoint of last interval used to calculate the value of the chi-square statistic.
OBSGTR	observed percent of data greater than upper specification limit
OBSLSS	observed percent of data less than the lower specification limit
PCHISQ	p-value for chi-square goodness-of-fit test
SCALE	value of scale parameter for fitted distribution. For the normal distribution, this is either the value of σ specified with the SIGMA= option or the sample standard deviation. For all other distributions, this is either the value specified with the SCALE= option or the value estimated by the procedure.
SHAPE1	value of shape parameter for fitted distribution. For distributions without a shape parameter (normal and exponential distributions), _SHAPE1_ is set to missing. For the gamma, lognormal, and Weibull distributions, the value of _SHAPE1_ is either the value specified with the SHAPE= option or the value estimated by the procedure. For the beta distribution, _SHAPE1_ is either the value of α specified with the ALPHA= option or the value estimated by the procedure.
SHAPE2	value of shape parameter for fitted distribution. For the beta distribution, _SHAPE2_ is either the value of β specified with the BETA= option or the value estimated by the procedure. For all other distributions, _SHAPE2_ is set to missing.
TARGET	target value
USL	upper specification limit
VAR	variable name
WIDTH	width of histogram interval

OUTHISTOGRAM= Data Sets

The OUTHISTOGRAM= data set contains information about histogram intervals. Since you can specify multiple HISTOGRAM statements with the CAPABILITY procedure, you can create multiple OUTHISTOGRAM= data sets.

The data set contains a group of observations for each variable plotted with the HISTOGRAM statement. The group contains an observation for each interval of the histogram, beginning with the leftmost interval that contains a value of the variable and ending with the rightmost interval that contains a value of the variable. These intervals will not necessarily coincide with the intervals displayed in the histogram since the histogram may be padded with empty intervals at either end. If you superimpose one or more fitted curves on the histogram, the OUTHISTOGRAM= data set contains multiple groups of observations for each variable (one group for each curve).

If you use a BY statement, the OUTHISTOGRAM= data set contains groups of observations for each BY group. ID variables are not saved in the OUTHISTOGRAM= data set.

The OUTHISTOGRAM= data set contains the variables listed in Table 11.19. By default, an OUTHISTOGRAM= data set contains the _MIDPT_ variable, whose values identify histogram intervals by their midpoints. When the ENDPOINTS= or NENDPOINTS option is specified, intervals are identified by endpoint values instead. If the RTINCLUDE option is specified, the _MAXPT_ variable contains an interval's upper endpoint value. Otherwise, the _MINPT_ variable contains the interval's lower endpoint value.

Table 11.19. Variables in the OUTHISTOGRAM= Data Set

Variable	Description
CURVE	name of fitted distribution (if requested in HISTOGRAM statement)
EXPPCT	estimated percent of population in histogram interval determined from optional fitted distribution
MAXPT	upper endpoint of histogram interval
MIDPT	midpoint of histogram interval
MINPT	lower endpoint of histogram interval
OBSPCT	percent of variable values in histogram interval
VAR	variable name

ODS Tables

The following table summarizes the ODS tables related to fitted distributions that you can request with the HISTOGRAM statement.

Table 11.20. ODS Tables Produced with the HISTOGRAM Statement

Table Name	Description	Option
Bins	histogram bins	MIDPERCENTS sub-option with any distribution option, such as NORMAL(MIDPERCENTS)
FitIndices	capability indices computed from fitted distribution	INDICES sub-option with any distribution option, such as LOGNORMAL(INDICES)
FitQuantiles	quantiles of fitted distribution	any distribution option such as NORMAL
GoodnessOfFit	goodness-of-fit tests for fitted distribution	any distribution option such as NORMAL
ParameterEstimates	parameter estimates for fitted distribution	any distribution option such as NORMAL
Specifications	percents outside specification limits based on empirical and fitted distributions	any distribution option such as NORMAL

SYMBOL and PATTERN Statement Options

In earlier releases of SAS/QC software, graphical features (such as colors and line types) of specification lines, histogram bars, and fitted curves were controlled with options in SYMBOL and PATTERN statements. These options are still supported, although they have been superseded by options in the HISTOGRAM and SPEC statements. The following tables summarize the two sets of options.

Table 11.21. Graphical Enhancement of Histogram Outlines and Specification Lines

Feature	Statement and Options	Alternative Statement and Options
Outline of Histogram Bars	HISTOGRAM Statement	SYMBOL1 Statement
color	CBARLINE=*color*	C=*color*
width		W=*value*
Target Reference Line	SPEC Statement	SYMBOL1 Statement
position	TARGET=*value*	
color	CTARGET=*color*	C=*color*
line type	LTARGET=*linetype*	L=*linetype*
width	WTARGET=*value*	W=*value*
Lower Specification Line	SPEC Statement	SYMBOL2 Statement
position	LSL=*value*	
color	CLSL=*color*	C=*color*
line type	LLSL=*linetype*	L=*linetype*
width	WLSL=*value*	W=*value*
Upper Specification Line	SPEC Statement	SYMBOL3 Statement
position	USL=*value*	
color	CUSL=*color*	C=*color*
line type	LUSL=*linetype*	L=*linetype*
width	WUSL=*value*	W=*value*

Table 11.22. Graphical Enhancement of Areas Under Histograms and Curves

Area Under Histogram or Curve	Statement and Options	Alternative Statement and Options
Histogram or Curve	HISTOGRAM Statement	PATTERN1 Statement
pattern	PFILL=*pattern*	V=*pattern*
color	CFILL=*color*	C=*color*
Left of Lower Specification Limit	SPEC Statement	PATTERN2 Statement
pattern	PLEFT=*pattern*	V=*pattern*
color	CLEFT=*color*	C=*color*
Right of Upper Specification Limit	SPEC Statement	PATTERN3 Statement
pattern	PRIGHT=*pattern*	V=*pattern*
color	CRIGHT=*color*	C=*color*

Table 11.23. Graphical Enhancement of Fitted Curves

Feature	Statement and Options	Alternative Statement and Options
Normal Curve	Normal-options	SYMBOL4 Statement
color	COLOR=*color*	C=*color*
line type	L=*linetype*	L=*linetype*
width	W=*value*	W=*value*
Lognormal Curve	Lognormal-options	SYMBOL5 Statement
color	COLOR=*color*	C=*color*
line type	L=*linetype*	L=*linetype*
width	W=*value*	W=*value*
Exponential Curve	Exponential-options	SYMBOL6 Statement
color	COLOR=*color*	C=*color*
line type	L=*linetype*	L=*linetype*
width	W=*value*	W=*value*
Weibull Curve	Weibull-options	SYMBOL7 Statement
color	COLOR=*color*	C=*color*
line type	L=*linetype*	L=*linetype*
width	W=*value*	W=*value*
Gamma Curve	Gamma-options	SYMBOL8 Statement
color	COLOR=*color*	C=*color*
line type	L=*linetype*	L=*linetype*
width	W=*value*	W=*value*
Beta Curve	Beta-options	SYMBOL9 Statement
color	COLOR=*color*	C=*color*
line type	L=*linetype*	L=*linetype*
width	W=*value*	W=*value*
S_B Curve	S_B-options	SYMBOL10 Statement
color	COLOR=*color*	C=*color*
line type	L=*linetype*	L=*linetype*
width	W=*value*	W=*value*
S_U Curve	S_U-options	SYMBOL11 Statement
color	COLOR=*color*	C=*color*
line type	L=*linetype*	L=*linetype*
width	W=*value*	W=*value*

Examples

This section provides advanced examples of the HISTOGRAM statement.

Example 11.1. Fitting a Beta Curve

See CAPBTA2 in the SAS/QC Sample Library

You can use a beta distribution to model the distribution of a quantity that is known to vary between lower and upper bounds. In this example, a manufacturing company uses a robotic arm to attach hinges on metal sheets. The attachment point should be offset 10.1 mm from the left edge of the sheet. The actual offset varies between 10.0

and 10.5 mm due to variation in the arm. Offsets for 50 attachment points are saved in the following data set:

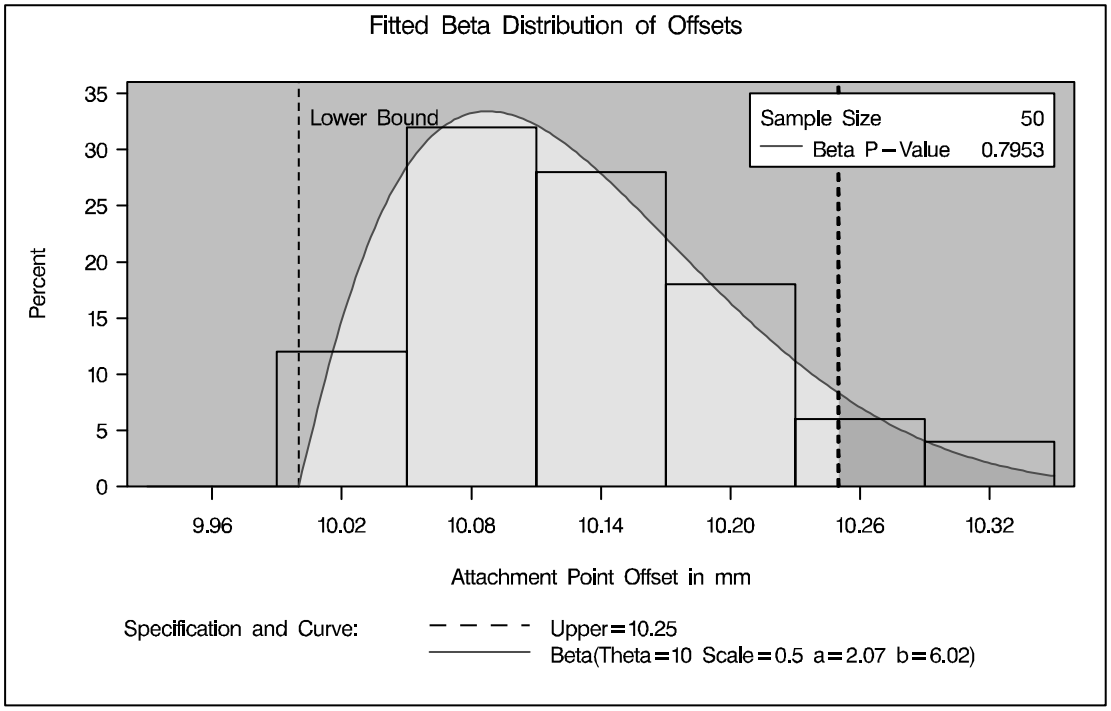
```
title ;
data measures;
   input length @@;
   label length = 'Attachment Point Offset in mm';
   datalines;
10.147 10.070 10.032 10.042 10.102
10.034 10.143 10.278 10.114 10.127
10.122 10.018 10.271 10.293 10.136
10.240 10.205 10.186 10.186 10.080
10.158 10.114 10.018 10.201 10.065
10.061 10.133 10.153 10.201 10.109
10.122 10.139 10.090 10.136 10.066
10.074 10.175 10.052 10.059 10.077
10.211 10.122 10.031 10.322 10.187
10.094 10.067 10.094 10.051 10.174
;
run;
```

The following statements create a histogram with a fitted beta density curve:

```
title1 'Fitted Beta Distribution of Offsets';
proc capability data=measures;
   specs usl = 10.25 lusl = 20 cusl = black
         cright = orange;
   histogram length /
      beta(theta=10 scale=0.5 color=red fill)
      cfill     = yellow
      cframe    = ligr
      href      = 10
      hreflabel = 'Lower Bound'
      lhref     = 2
      vaxis     = axis1;
   axis1 label=(a=90 r=0);
   inset n = 'Sample Size'
         beta(pchisq = 'P-Value') / pos=ne  cfill=blank;
run;
```

The histogram is shown in Output 11.1.1. The THETA= *beta-option* specifies the lower threshold. The SCALE= *beta-option* specifies the range between the lower threshold and the upper threshold (in this case, 0.5 mm). Note that in general, the default THETA= and SCALE= values are zero and one, respectively.

Output 11.1.1. Superimposing a Histogram with a Fitted Beta Curve



The FILL *beta-option* specifies that the area under the curve is to be filled with the CFILL= color. (If FILL were omitted, the CFILL= color would be used to fill the histogram bars instead.) The CRIGHT= option in the SPEC statement specifies the color under the curve to the right of the upper specification limit. If the CRIGHT= option were not specified, the entire area under the curve would be filled with the CFILL= color. When a lower specification limit is available, you can use the CLEFT= option in the SPEC statement to specify the color under the curve to the left of this limit.

The HREF= option draws a reference line at the lower bound, and the HREFLABEL= option adds the label *Lower Bound*. The option LHREF=2 specifies a dashed line type. The INSET statement adds an inset with the sample size and the p-value for a chi-square goodness-of-fit test.

In addition to displaying the beta curve, the BETA option summarizes the curve fit, as shown in Output 11.1.2. The output tabulates the parameters for the curve, the chi-square goodness-of-fit test whose p-value is shown in Output 11.1.1, the observed and estimated percents above the upper specification limit, and the observed and estimated quantiles. For instance, based on the beta model, the percent of offsets greater than the upper specification limit is 6.6%. For computational details, see "Formulas for Fitted Curves" on page 313.

Output 11.1.2. Summary of Fitted Beta Distribution

```
                Fitted Beta Distribution of Offsets

          Variable:  length  (Attachment Point Offset in mm)

                        Specification Limits

            --------Limit--------      ------Percent-------

            Lower (LSL)                % < LSL
            Target                     % Between
            Upper (USL)   10.25000     % > USL      8.000000


                Fitted Beta Distribution of Offsets

                Fitted Beta Distribution for length

                  Parameters for Beta Distribution

                   Parameter    Symbol    Estimate

                   Threshold    Theta           10
                   Scale        Sigma          0.5
                   Shape        Alpha      2.06832
                   Shape        Beta       6.022479
                   Mean                    10.12782
                   Std Dev                 0.072339


             Goodness-of-Fit Tests for Beta Distribution

Test                   ----Statistic-----      DF    ------p Value------

Chi-Square             Chi-Sq  1.02463588       3    Pr > Chi-Sq    0.795


                    Quantiles for Beta Distribution

                              ------Quantile------
                    Percent   Observed   Estimated

                        1.0    10.0180     10.0124
                        5.0    10.0310     10.0285
                       10.0    10.0380     10.0416
                       25.0    10.0670     10.0718
                       50.0    10.1220     10.1174
                       75.0    10.1750     10.1735
                       90.0    10.2255     10.2292
                       95.0    10.2780     10.2630
                       99.0    10.3220     10.3237
```

Example 11.2. Fitting Lognormal, Weibull, and Gamma Curves

See CAPCURV in the SAS/QC Sample Library

To find an appropriate model for a process distribution, you should consider curves from several distribution families. As shown in this example, you can use the HISTOGRAM statement to fit more than one type of distribution and display the density curves on the same histogram.

The gap between two plates is measured (in cm) for each of 50 welded assemblies selected at random from the output of a welding process assumed to be in statistical control. The lower and upper specification limits for the gap are 0.3 cm and 0.8 cm, respectively. The measurements are saved in a data set named PLATES.

```
data plates;
   label gap='Plate Gap in cm';
   input gap @@;
   datalines;
0.746  0.357  0.376  0.327  0.485 1.741  0.241  0.777  0.768  0.409
0.252  0.512  0.534  1.656  0.742 0.378  0.714  1.121  0.597  0.231
0.541  0.805  0.682  0.418  0.506 0.501  0.247  0.922  0.880  0.344
0.519  1.302  0.275  0.601  0.388 0.450  0.845  0.319  0.486  0.529
1.547  0.690  0.676  0.314  0.736 0.643  0.483  0.352  0.636  1.080
;
run;
```

The following statements fit three distributions (lognormal, Weibull, and gamma) and display their density curves on a single histogram:

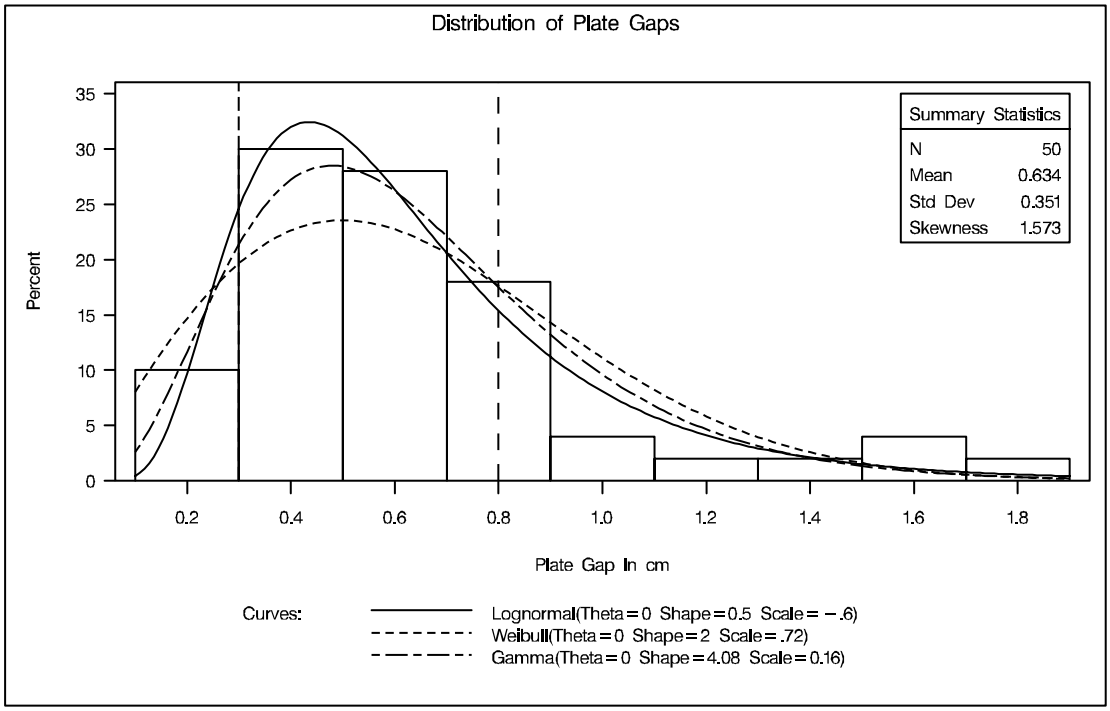
```
title1 'Distribution of Plate Gaps';
proc capability data=plates;
   var gap;
   specs  lsl = 0.3  llsl = 3 usl = 0.8  lusl = 20;
   histogram /
       midpoints=0.2 to 1.8 by 0.2
       lognormal (l=1)
       weibull   (l=2)
       gamma     (l=8)
       nospeclegend
       vaxis   =  axis1;
   inset n mean(5.3) std='Std Dev'(5.3) skewness(5.3)
          / pos = ne  header = 'Summary Statistics';
   axis1 label=(a=90 r=0);
run;
```

The LOGNORMAL, WEIBULL, and GAMMA options superimpose fitted curves on the histogram in Output 11.2.1. The L= options specify distinct line types for the curves. Note that a threshold parameter $\theta = 0$ is assumed for each curve. In applications where the threshold is not zero, you can specify θ with the THETA= option.

The LOGNORMAL, WEIBULL, and GAMMA options also produce the summaries for the fitted distributions shown in Output 11.2.2, Output 11.2.3, and Output 11.2.4.

Output 11.2.2 provides four goodness-of-fit tests for the lognormal distribution: the chi-square test and three tests based on the EDF (Anderson-Darling, Cramer-von Mises, and Kolmogorov-Smirnov). See "Chi-Square Goodness-of-Fit Test" on page 323 and "EDF Goodness-of-Fit Tests" on page 323 for more information. The EDF tests are superior to the chi-square test because they are not dependent on the set of midpoints used for the histogram.

Output 11.2.1. Superimposing a Histogram with Fitted Curves



At the $\alpha = 0.10$ significance level, all four tests support the conclusion that the two-parameter lognormal distribution with scale parameter $\hat{\zeta} = -0.58$, and shape parameter $\hat{\sigma} = 0.50$ provides a good model for the distribution of plate gaps.

Output 11.2.3 provides two EDF goodness-of-fit tests for the Weibull distribution: the Anderson-Darling and the Cramer-von Mises tests. (See Table 11.17 on page 326 for a complete list of the EDF tests available in the HISTOGRAM statement.) The probability values for the chi-square and EDF tests are all less than 0.10, indicating that the data do not support a Weibull model.

Output 11.2.4 provides a chi-square goodness-of-fit test for the gamma distribution. (None of the EDF tests are currently supported when the scale and shape parameter of the gamma distribution are estimated; see Table 11.17 on page 326.) The probability value for the chi-square test is less than 0.10, indicating that the data do not support a gamma model.

Based on this analysis, the fitted lognormal distribution is the best model for the distribution of plate gaps. You can use this distribution to calculate useful quantities. For instance, you can compute the probability that the gap of a randomly sampled plate exceeds the upper specification limit, as follows:

$$\begin{aligned} \Pr[\text{gap} > \text{USL}] &= \Pr\left[Z > \tfrac{1}{\sigma}(\log(\text{USL} - \theta) - \zeta)\right] \\ &= 1 - \Phi\left[\tfrac{1}{\sigma}(\log(\text{USL} - \theta) - \zeta)\right] \end{aligned}$$

where Z has a standard normal distribution, and $\Phi(\cdot)$ is the standard normal cumulative distribution function. Note that $\Phi(\cdot)$ can be computed with the DATA step

Output 11.2.2. Summary of Fitted Lognormal Distribution

Distribution of Plate Gaps

Fitted Lognormal Distribution for gap

Parameters for Lognormal Distribution

Parameter	Symbol	Estimate
Threshold	Theta	0
Scale	Zeta	-0.58375
Shape	Sigma	0.499546
Mean		0.631932
Std Dev		0.336436

Goodness-of-Fit Tests for Lognormal Distribution

Test	----Statistic-----		DF	------p Value------	
Kolmogorov-Smirnov	D	0.06441431		Pr > D	>0.150
Cramer-von Mises	W-Sq	0.02823022		Pr > W-Sq	>0.500
Anderson-Darling	A-Sq	0.24308402		Pr > A-Sq	>0.500
Chi-Square	Chi-Sq	7.51762213	6	Pr > Chi-Sq	0.276

Percent Outside Specifications for Lognormal Distribution

Lower Limit		Upper Limit	
LSL	0.300000	USL	0.800000
Obs Pct < LSL	10.000000	Obs Pct > USL	20.000000
Est Pct < LSL	10.719540	Est Pct > USL	23.519008

Quantiles for Lognormal Distribution

	------Quantile------	
Percent	Observed	Estimated
1.0	0.23100	0.17449
5.0	0.24700	0.24526
10.0	0.29450	0.29407
25.0	0.37800	0.39825
50.0	0.53150	0.55780
75.0	0.74600	0.78129
90.0	1.10050	1.05807
95.0	1.54700	1.26862
99.0	1.74100	1.78313

Output 11.2.3. Summary of Fitted Weibull Distribution

Distribution of Plate Gaps

Fitted Weibull Distribution for gap

Parameters for Weibull Distribution

Parameter	Symbol	Estimate
Threshold	Theta	0
Scale	Sigma	0.719208
Shape	C	1.961159
Mean		0.637641
Std Dev		0.339248

Goodness-of-Fit Tests for Weibull Distribution

Test	----Statistic-----		DF	------p Value------	
Cramer-von Mises	W-Sq	0.1593728		Pr > W-Sq	0.016
Anderson-Darling	A-Sq	1.1569354		Pr > A-Sq	<0.010
Chi-Square	Chi-Sq	15.0252996	6	Pr > Chi-Sq	0.020

Percent Outside Specifications for Weibull Distribution

Lower Limit		Upper Limit	
LSL	0.300000	USL	0.800000
Obs Pct < LSL	10.000000	Obs Pct > USL	20.000000
Est Pct < LSL	16.473319	Est Pct > USL	29.165543

Quantiles for Weibull Distribution

	------Quantile------	
Percent	Observed	Estimated
1.0	0.23100	0.06889
5.0	0.24700	0.15817
10.0	0.29450	0.22831
25.0	0.37800	0.38102
50.0	0.53150	0.59661
75.0	0.74600	0.84955
90.0	1.10050	1.10040
95.0	1.54700	1.25842
99.0	1.74100	1.56691

Output 11.2.4. Summary of Fitted Gamma Distribution

Distribution of Plate Gaps

Fitted Gamma Distribution for gap

Parameters for Gamma Distribution

Parameter	Symbol	Estimate
Threshold	Theta	0
Scale	Sigma	0.155198
Shape	Alpha	4.082646
Mean		0.63362
Std Dev		0.313587

Goodness-of-Fit Tests for Gamma Distribution

Test	----Statistic-----		DF	------p Value------	
Kolmogorov-Smirnov	D	0.0969533		Pr > D	>0.250
Cramer-von Mises	W-Sq	0.0739847		Pr > W-Sq	>0.250
Anderson-Darling	A-Sq	0.5810661		Pr > A-Sq	0.137
Chi-Square	Chi-Sq	12.3075959	6	Pr > Chi-Sq	0.055

Percent Outside Specifications for Gamma Distribution

Lower Limit		Upper Limit	
LSL	0.300000	USL	0.800000
Obs Pct < LSL	10.000000	Obs Pct > USL	20.000000
Est Pct < LSL	12.111039	Est Pct > USL	25.696522

Quantiles for Gamma Distribution

	------Quantile------	
Percent	Observed	Estimated
1.0	0.23100	0.13326
5.0	0.24700	0.21951
10.0	0.29450	0.27938
25.0	0.37800	0.40404
50.0	0.53150	0.58271
75.0	0.74600	0.80804
90.0	1.10050	1.05392
95.0	1.54700	1.22160
99.0	1.74100	1.57939

function PROBNORM. In this example, USL $= 0.8$ and $\Pr[\text{gap} > 0.8] = 0.2352$. This value is expressed as a percent (*Est Pct* > *USL*) in Output 11.2.2.

Example 11.3. Comparing Goodness-of-Fit Tests

See CAPGOF in the SAS/QC Sample Library

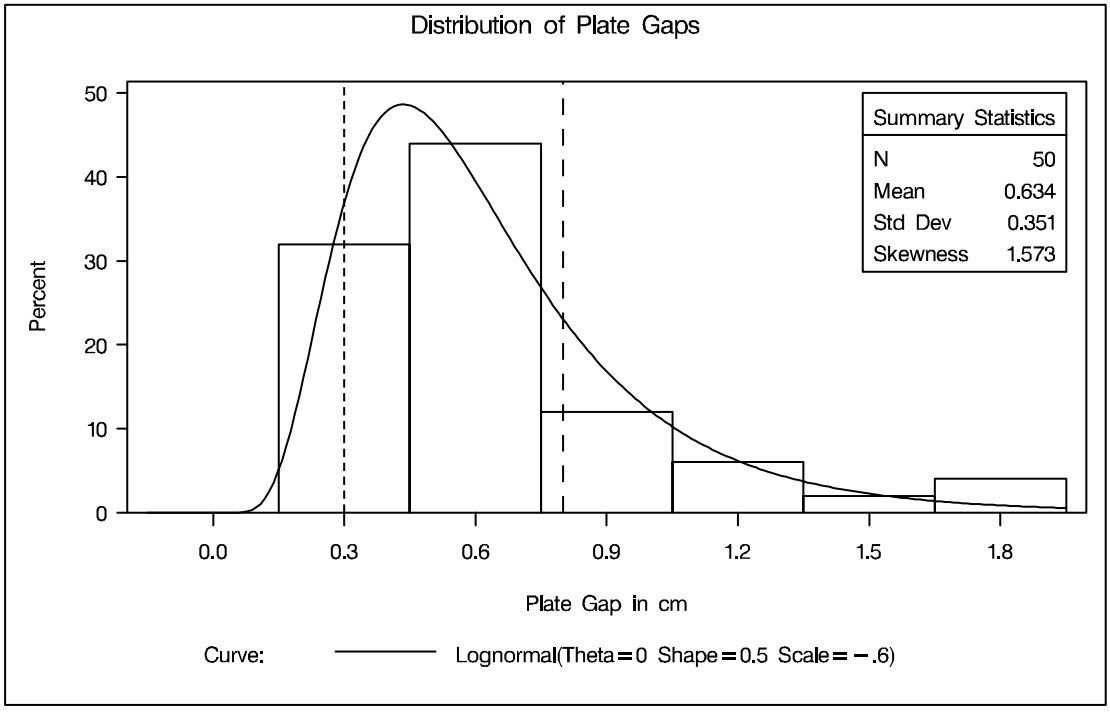
A weakness of the chi-square goodness-of-fit test is its dependence on the choice of histogram midpoints. An advantage of the EDF tests is that they give the same results regardless of the midpoints, as illustrated in this example.

In Example 11.2, the option MIDPOINTS=0.2 TO 1.8 BY 0.2 was used to specify the histogram midpoints for GAP. The following statements refit the lognormal distribution using default midpoints (0.3 to 1.8 by 0.3).

```
title1 'Distribution of Plate Gaps';
proc capability data=plates noprint;
   var gap;
   specs  lsl = 0.3 llsl = 2 usl = 0.8 lusl = 20;
   histogram /
       lognormal (l=1)
       nospeclegend
       vaxis  = axis1;
   inset n mean(5.3) std='Std Dev'(5.3) skewness(5.3)
       / pos = ne  header = 'Summary Statistics';
   axis1 label=(a=90 r=0);
run;
```

The histogram is shown in Output 11.3.1.

Output 11.3.1. Lognormal Curve Fit with Default Midpoints



A summary of the lognormal fit is shown in Output 11.3.2. The p-value for the chi-square goodness-of-fit test is 0.0822. Since this value is less than 0.10 (a typical cutoff level), the conclusion is that the lognormal distribution is not an appropriate model for the data. This is the *opposite* conclusion drawn from the chi-square test in Example 11.2, which is based on a different set of midpoints and has a p-value of 0.2756 (see Output 11.2.2). Moreover, the results of the EDF goodness-of-fit tests are the same since these tests do not depend on the midpoints. When available, the EDF tests provide more powerful alternatives to the chi-square test. For a thorough discussion of EDF tests, refer to D'Agostino and Stephens (1986).

Output 11.3.2. Printed Output for the Lognormal Curve

Distribution of Plate Gaps

Fitted Lognormal Distribution for gap

Parameters for Lognormal Distribution

Parameter	Symbol	Estimate
Threshold	Theta	0
Scale	Zeta	-0.58375
Shape	Sigma	0.499546
Mean		0.631932
Std Dev		0.336436

Goodness-of-Fit Tests for Lognormal Distribution

Test	----Statistic-----		DF	------p Value------	
Kolmogorov-Smirnov	D	0.06441431		Pr > D	>0.150
Cramer-von Mises	W-Sq	0.02823022		Pr > W-Sq	>0.500
Anderson-Darling	A-Sq	0.24308402		Pr > A-Sq	>0.500
Chi-Square	Chi-Sq	6.69789360	3	Pr > Chi-Sq	0.082

Example 11.4. Computing Capability Indices for Nonnormal Distributions

See CAPIND in the SAS/QC Sample Library

Standard capability indices such as C_{pk} are generally considered meaningful only if the process output has a normal (or reasonably normal) distribution. In practice, however, many processes have nonnormal distributions. This example, which is a continuation of Example 11.2 and Example 11.3, shows how you can use the HISTOGRAM statement to compute generalized capability indices based on fitted nonnormal distributions.

The following statements produce printed output that is partially listed in Output 11.4.1 and Output 11.4.2:

```
proc capability data=plates;
   specs lsl=0.3 usl= 0.8 alpha=0.05;
   histogram gap / lognormal(indices) noplot;
run;
```

The PROC CAPABILITY statement computes the standard capability indices that are shown in Output 11.4.1.

Output 11.4.1. Standard Capability Indices for Variable GAP

```
                        The CAPABILITY Procedure
                    Variable:  gap  (Plate Gap in cm)

                       Process Capability Indices

               Index       Value     95% Confidence Limits

               Cp       0.237112     0.190279     0.283853
               CPL      0.316422     0.203760     0.426833
               CPU      0.157803     0.059572     0.254586
               Cpk      0.157803     0.060270     0.255336

Warning:  Normality is rejected for alpha = 0.05 using the Shapiro-Wilk test
```

The ALPHA= option in the SPECS statement requests a Kolmogorov-Smirnov goodness-of-fit test for normality in conjunction with the indices and displays the warning that normality is rejected at the significance level $\alpha = 0.05$.

Example 11.2 concluded that the fitted lognormal distribution summarized in Output 11.2.2 is a good model, so one might consider computing generalized capability indices based on this distribution. These indices are requested with the INDICES option and are shown in Output 11.4.2. Formulas and recommendations for these indices are given in "Indices Using Fitted Curves" on page 325.

Output 11.4.2. Fitted Lognormal Distribution Information

```
                  Fitted Lognormal Distribution for gap

            Capability Indices Based on Lognormal Distribution

                                Cp       0.210804
                                CPL      0.595156
                                CPU      0.124927
                                Cpk      0.124927
```

Example 11.5. Computing Kernel Density Estimates

See CAPKERN1 in the SAS/QC Sample Library

This example illustrates the use of kernel density estimates to visualize a nonnormal data distribution.

The effective channel length (in microns) is measured for 1225 field effect transistors. The channel lengths are saved as values of the variable LENGTH in a SAS data set named CHANNEL:

```
data channel;
   length lot $ 16;
   input length @@;
   select;
        when (_n_ <= 425) Lot='Lot 1';
        when (_n_ >= 926) Lot='Lot 3';
        otherwise Lot='Lot 2';
     end;
   datalines;
0.91 1.01 0.95 1.13 1.12 0.86 0.96 1.17 1.36 1.10
0.98 1.27 1.13 0.92 1.15 1.26 1.14 0.88 1.03 1.00
0.98 0.94 1.09 0.92 1.10 0.95 1.05 1.05 1.11 1.15
1.11 0.98 0.78 1.09 0.94 1.05 0.89 1.16 0.88 1.19
1.01 1.08 1.19 0.94 0.92 1.27 0.90 0.88 1.38 1.02

  ...

1.80 2.35 2.23 1.96 2.16 2.08 2.06 2.03 2.18 1.83
2.13 2.05 1.90 2.07 2.15 1.96 2.15 1.89 2.15 2.04
1.95 1.93 2.22 1.74 1.91
;
```

When you use kernel density estimates to explore a data distribution, you should try several choices for the bandwidth parameter c since this determines the smoothness and closeness of the fit. You can specify a list of C= values with the KERNEL option to request multiple density estimates, as shown in the following statements:

```
title "FET Channel Length Analysis";
proc capability data=channel noprint;
   histogram length / kernel(c = 0.25 0.50 0.75 1.00
                             l = 1 20 2 34
                             color=red);
run;
```

The L= option specifies distinct line types for the curves (the L= values are paired with the C= values in the order listed). The display, shown in Output 11.5.1, demonstrates the effect of c. In general, larger values of c yield smoother density estimates, and smaller values yield estimates that more closely fit the data distribution.

Output 11.5.1 reveals strong trimodality in the data, which are explored further in "Creating a One-Way Comparative Histogram" on page 248.

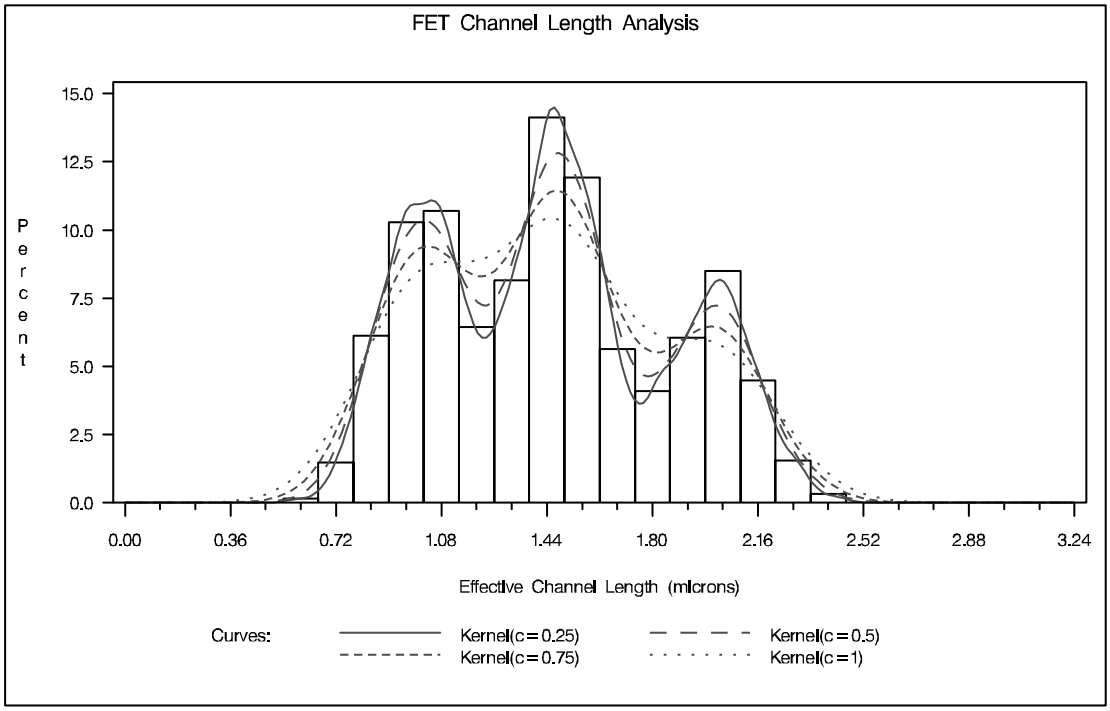
Example 11.6. Fitting a Three-Parameter Lognormal Curve

See CAPL3A in the SAS/QC Sample Library

If you request a lognormal fit with the LOGNORMAL option, a *two-parameter* lognormal distribution is assumed. This means that the shape parameter σ and the scale parameter ζ are unknown (unless specified) and that the threshold θ is known (it is either specified with the THETA= option or assumed to be zero).

If it is necessary to estimate θ in addition to ζ and σ, the distribution is referred to as a *three-parameter* lognormal distribution. The equation for this distribution is the

Output 11.5.1. Multiple Kernel Density Estimates



same as the equation given on page 318, but the method of maximum likelihood must be modified. This example shows how you can request a three-parameter lognormal distribution.

A manufacturing process (assumed to be in statistical control) produces a plastic laminate whose strength must exceed a minimum of 25 psi. Samples are tested, and a lognormal distribution is observed for the strengths. It is important to estimate θ to determine whether the process is capable of meeting the strength requirement. The strengths for 49 samples are saved in the following data set:

```
data plastic;
   label strength='Strength in psi';
   input strength @@;
   datalines;
30.26 31.23 71.96 47.39 33.93 76.15 42.21
81.37 78.48 72.65 61.63 34.90 24.83 68.93
43.27 41.76 57.24 23.80 34.03 33.38 21.87
31.29 32.48 51.54 44.06 42.66 47.98 33.73
25.80 29.95 60.89 55.33 39.44 34.50 73.51
43.41 54.67 99.43 50.76 48.81 31.86 33.88
35.57 60.41 54.92 35.66 59.30 41.96 45.32
;
run;
```

The following statements use the LOGNORMAL option in the HISTOGRAM statement to display the fitted three-parameter lognormal curve shown in Output 11.6.1:

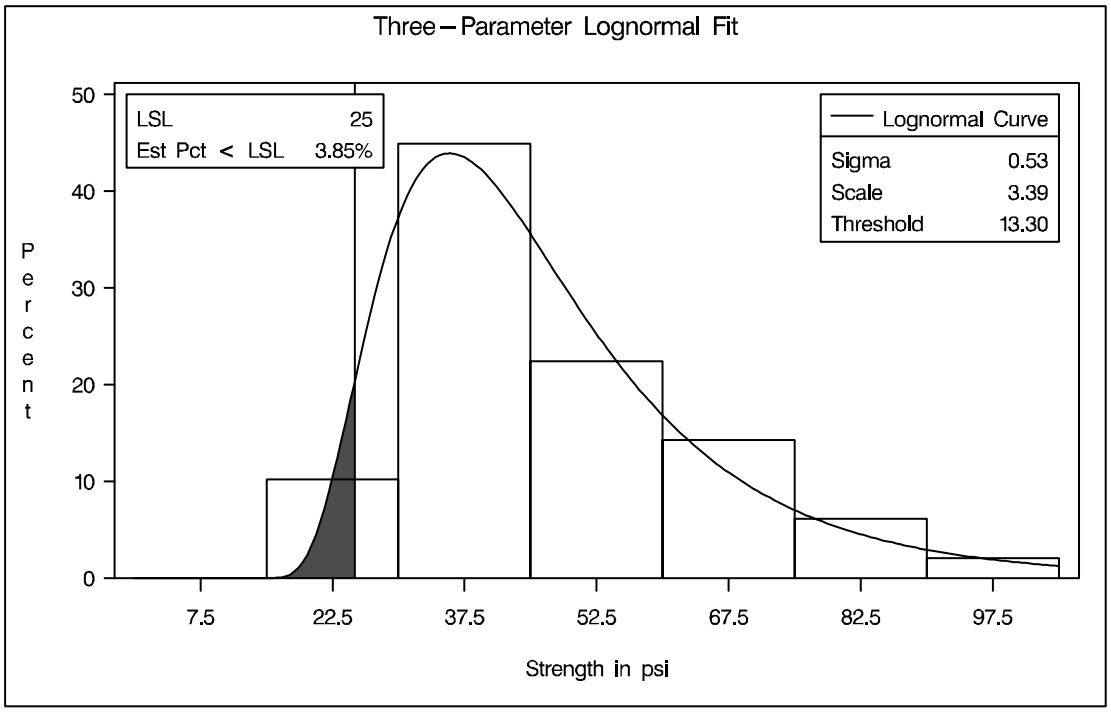
```
title 'Three-Parameter Lognormal Fit';
proc capability data=plastic noprint;
   spec lsl=25 cleft=green;
   histogram strength / lognormal(fill theta = est)
                        cfill = white
                        nolegend;
   inset lsl='LSL' lslpct / cfill=blank pos=nw;
   inset lognormal        / format=6.2 pos=ne;
run;
```

Specifying THETA=EST requests a *local* maximum likelihood estimate (LMLE) for θ, as described by Cohen (1951). This estimate is then used to compute maximum likelihood estimates for σ and ζ. The sample program CAPL3A illustrates a similar computational method implemented as a SAS/IML program.

Note that you can specify THETA=EST as a *Weibull-option* to fit a three-parameter Weibull distribution.

See CAPW3A in the SAS/QC Sample Library

Output 11.6.1. Three-Parameter Lognormal Fit



Example 11.7. Annotating a Folded Normal Curve

This example shows how to display a fitted curve that is not supported by the HISTOGRAM statement.

See FNORM2 in the SAS/QC Sample Library

The offset of an attachment point is measured (in mm) for a number of manufactured assemblies, and the measurements are saved in a data set named ASSEMBLY.

```
data assembly;
  label offset = 'Offset (in mm)';
  input offset @@;
  datalines;
11.11 13.07 11.42  3.92 11.08  5.40 11.22 14.69  6.27  9.76
 9.18  5.07  3.51 16.65 14.10  9.69 16.61  5.67  2.89  8.13
 9.97  3.28 13.03 13.78  3.13  9.53  4.58  7.94 13.51 11.43
11.98  3.90  7.67  4.32 12.69  6.17 11.48  2.82 20.42  1.01
 3.18  6.02  6.63  1.72  2.42 11.32 16.49  1.22  9.13  3.34
 1.29  1.70  0.65  2.62  2.04 11.08 18.85 11.94  8.34  2.07
 0.31  8.91 13.62 14.94  4.83 16.84  7.09  3.37  0.49 15.19
 5.16  4.14  1.92 12.70  1.97  2.10  9.38  3.18  4.18  7.22
15.84 10.85  2.35  1.93  9.19  1.39 11.40 12.20 16.07  9.23
 0.05  2.15  1.95  4.39  0.48 10.16  4.81  8.28  5.68 22.81
 0.23  0.38 12.71  0.06 10.11 18.38  5.53  9.36  9.32  3.63
12.93 10.39  2.05 15.49  8.12  9.52  7.77 10.70  6.37  1.91
 8.60 22.22  1.74  5.84 12.90 13.06  5.08  2.09  6.41  1.40
15.60  2.36  3.97  6.17  0.62  8.56  9.36 10.19  7.16  2.37
12.91  0.95  0.89  3.82  7.86  5.33 12.92  2.64  7.92 14.06
;
run;
```

The assembly process is in statistical control, and it is decided to fit a *folded normal distribution* to the offset measurements. A variable X has a folded normal distribution if $X = |Y|$, where Y is distributed as $N(\mu, \sigma)$. The fitted density is

$$h(x) = \frac{1}{\sqrt{2\pi}\sigma}\left[\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) + \exp\left(-\frac{(x+\mu)^2}{2\sigma^2}\right)\right] \quad , \; x \geq 0$$

You can use SAS/IML software to compute preliminary estimates of μ and σ based on a method of moments given by Elandt (1961). These estimates are computed by solving equation (19) of Elandt (1961), which is given by

$$f(\theta) = \frac{\left(\frac{2}{\sqrt{2\pi}}e^{-\theta^2/2} - \theta\left[1 - 2\Phi(\theta)\right]\right)^2}{1+\theta^2} = A$$

where $\Phi(\cdot)$ is the standard normal distribution function, and

$$A = \frac{\bar{x}^2}{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$$

Then the estimates of σ and μ are given by

$$\begin{aligned} \hat{\sigma}_0 &= \sqrt{\frac{\frac{1}{n}\sum_{i=1}^{n} x_i^2}{1+\hat{\theta}^2}} \\ \hat{\mu}_0 &= \hat{\theta} \cdot \hat{\sigma}_0 \end{aligned}$$

Begin by using the MEANS procedure to compute the first and second moments and using the DATA step to compute the constant A.

```
proc means data = assembly noprint;
   var offset;
   output out=stat mean=m1 var=var n=n min = min;
run;

* Compute constant A from equation (19) of Elandt (1961) ;
data stat;
   keep m2 a min;
   set stat;
   a  = (m1*m1);
   m2 = ((n-1)/n)*var + a;
   a  = a/m2;
run;
```

Next, use the SAS/IML subroutine NLPDD to solve equation (19) by minimizing $(f(\theta) - A)^2$, and compute $\hat{\mu}_0$ and $\hat{\sigma}_0$.

```
proc iml;
   use stat;
   read all var {m2}  into m2;
   read all var {a}   into a;
   read all var {min} into min;

   * f(t) is the function in equation (19) of Elandt (1961) ;
   start f(t) global(a);
     y = .39894*exp(-0.5*t*t);
     y = (2*y-(t*(1-2*probnorm(t))))**2/(1+t*t);
     y = (y-a)**2;
     return(y);
   finish;

   * Minimize (f(t)-A)**2 and estimate mu and sigma ;
   if ( min < 0 ) then do;
      print "Warning: Observations are not all nonnegative.";
      print "     The folded normal is inappropriate.";
      stop;
      end;
   if ( a < 0.637 ) then do;
      print "Warning: the folded normal may be inappropriate";
      end;
   opt = { 0 0 };
   con = { 1e-6 };
   x0  = { 2.0 };
   tc  = { . . . . . 1e-12 . . . . . .};
   call nlpdd(rc,etheta0,"f",x0,opt,con,tc);
   esig0 = sqrt(m2/(1+etheta0*etheta0));
   emu0  = etheta0*esig0;

   create prelim var {emu0 esig0 etheta0};
   append;
   close prelim;
```

The preliminary estimates are saved in the data set PRELIM, as shown in Output 11.7.1.

Output 11.7.1. Preliminary Estimates of μ, σ, and θ

The Data Set PRELIM

EMU0	ESIG0	ETHETA0
6.51735	6.54953	0.99509

Now, using $\hat{\mu}_0$ and $\hat{\sigma}_0$ as initial estimates, call the NLPDD subroutine to maximize the log likelihood, $l(\mu, \sigma)$, of the folded normal distribution, where, up to a constant,

$$l(\mu,\sigma) = -n\log\sigma + \sum_{i=1}^{n}\log\left[\exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right) + \exp\left(-\frac{(x_i+\mu)^2}{2\sigma^2}\right)\right]$$

```
* Define the log likelihood of the folded normal ;
start g(p) global(x);
   y = 0.0;
   do i = 1 to nrow(x);
      z = exp( (-0.5/p[2])*(x[i]-p[1])*(x[i]-p[1]) );
      z = z + exp( (-0.5/p[2])*(x[i]+p[1])*(x[i]+p[1]) );
      y = y + log(z);
   end;
   y = y - nrow(x)*log( sqrt( p[2] ) );
   return(y);
finish;

* Maximize the log likelihood with subroutine NLPDD ;
use assembly;
read all var {offset} into x;
esig0sq = esig0*esig0;
x0      = emu0||esig0sq;
opt     = { 1 0 };
con     = { . 0.0, .  .  };
call nlpdd(rc,xr,"g",x0,opt,con);
emu     = xr[1];
esig    = sqrt(xr[2]);
etheta  = emu/esig;

create parmest var{emu esig etheta};
append;
close parmest;
quit;
```

The data set PARMEST saves the maximum likelihood estimates $\hat{\mu}$ and $\hat{\sigma}$ (as well as $\hat{\mu}/\hat{\sigma}$), as shown in Output 11.7.2.

Output 11.7.2. Final Estimates of μ, σ, and θ

The Data Set PARMEST

EMU	ESIG	ETHETA
6.66761	6.39650	1.04239

To annotate the curve on a histogram, begin by computing the width and endpoints of the histogram intervals. The following statements save these values in an OUTFIT= data set called OUT. Note that a plot is not produced at this point.

```
proc capability data = assembly noprint;
   histogram offset / outfit = out normal(noprint) noplot;
run;
```

Output 11.7.3 provides a partial listing of the data set OUT. The width and endpoints of the histogram bars are saved as values of the variables _WIDTH_, _MIDPT1_, and _MIDPTN_. See "Output Data Sets" on page 328.

Output 11.7.3. The OUTFIT= Data Set OUT

OUTFIT= Data Set OUT

VAR	_CURVE_	_LOCATN_	_SCALE_	_CHISQ_	_DF_	_PCHISQ_	_MIDPT1_	_WIDTH_
offset	NORMAL	7.62	5.24	31.17	5	0	1.5	3

MIDPTN	_EXPECT_	_ESTSTD_	_ADASQ_	_ADP_	_CVMWSQ_	_CVMP_	_KSD_	_KSP_
22.5	7.62	5.24	1.9	0.01	0.28	0.01	0.09	0.01

The following statements create an annotate data set named ANNO, which contains the coordinates of the fitted curve:

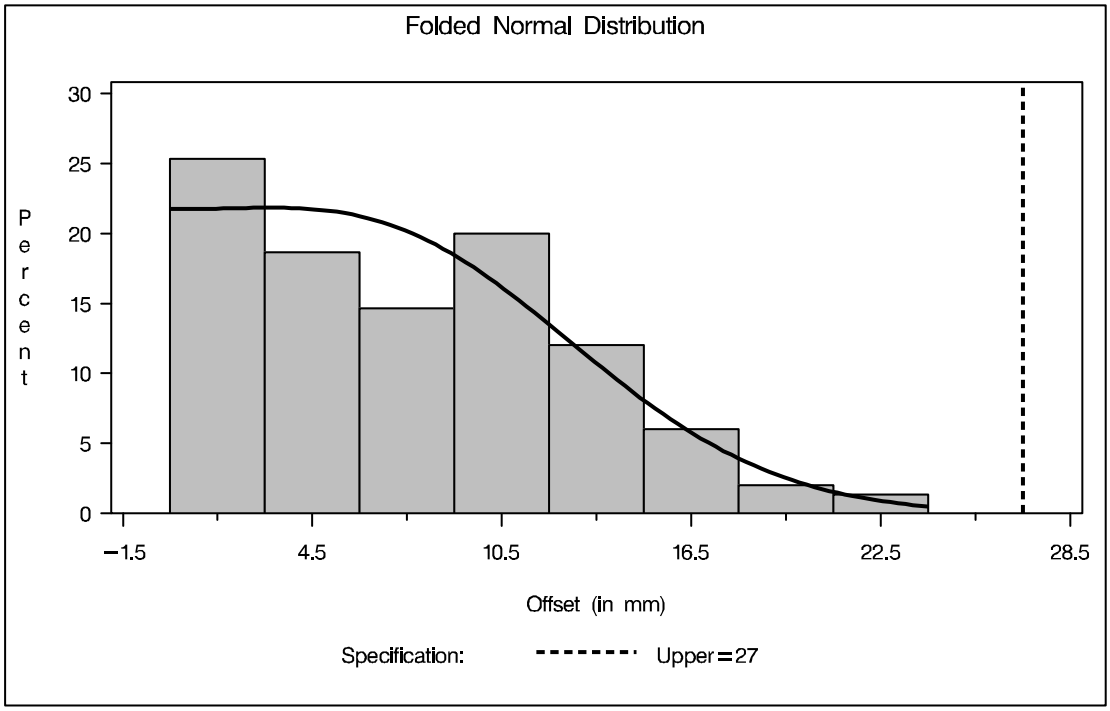
```
data anno;
   merge parmest out;
   length function color $ 8;

   function = 'point';
   color    = 'black';
   size     =  2;
   xsys     = '2';
   ysys     = '2';
   when     = 'a';
   constant = 39.894*_width_;;
  left     =  _midpt1_ - .5*_width_;
   right    =  _midptn_ + .5*_width_;
   inc      = (right-left)/100;
   do x = left to right by inc;
      z1 = (x-emu)/esig;
```

```
      z2 = (x+emu)/esig;
      y  = (constant/esig)*(exp(-0.5*z1*z1)+exp(-0.5*z2*z2));
      output;
      function = 'draw';
   end;
run;
```

The following statements read the ANNOTATE= data set and display the histogram and fitted curve, as shown in Output 11.7.4:

```
title "Folded Normal Distribution";
proc capability data=assembly noprint;
   spec usl=27 cusl=black lusl=2 wusl=2;
   histogram offset / annotate = anno
                      cbarline = black
                      cfill    = ligr;
run;
```

Output 11.7.4. Histogram with Annotated Folded Normal Curve



Chapter 12
INSET Statement

Chapter Contents

OVERVIEW 355

GETTING STARTED 355
- Displaying Summary Statistics on a Histogram 355
- Formatting Values and Customizing Labels 357
- Adding a Header and Positioning the Inset 358

SYNTAX 359
- Summary of INSET Keywords 362
- Summary of Options 367
- Dictionary of Options 368

DETAILS 370
- Positioning the Inset Using Compass Points 370
- Positioning the Inset in the Margins 371
- Positioning the Inset Using Coordinates 372

EXAMPLES 374
- Example 12.1. Inset for Goodness-of-Fit Statistics 374
- Example 12.2. Inset for Areas Under a Fitted Curve 375

Chapter 12
INSET Statement

Overview

Graphical displays such as histograms and probability plots are commonly used for process capability analysis. You can use the INSET statement to enhance these plots by adding a box or table (referred to as an *inset*) of summary statistics directly to the graph. An inset typically displays statistics calculated by the CAPABILITY procedure but can also display values provided in a SAS data set. A typical application of the INSET statement is to augment a histogram with the sample size, mean, standard deviation, and process capability index C_{pk}.

Note that the INSET statement by itself does not produce a display and must be used with the CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PPPLOT, PROBPLOT, or QQPLOT statement. *

You can use options in the INSET statement to

- specify the position of the inset
- specify a header for the inset table
- specify graphical enhancements, such as background colors, text colors, text height, text font, and drop shadows

Getting Started

This section introduces the INSET statement with examples that illustrate commonly used options. Complete syntax for the INSET statement is presented in the "Syntax" section on page 359, and advanced examples are given in the "Examples" section on page 374.

Displaying Summary Statistics on a Histogram

In a plant producing copper wire, an important quality characteristic is the torsion strength, measured as the twisting force in pounds per inch necessary to break the wire. The following statements create the SAS data set WIRE, which contains the torsion strengths (STRENGTH) for 50 different wire samples:

See CAPINS1 in the SAS/QC Sample Library

*In Release 6.12 and in previous releases of SAS/QC software, the keyword GRAPHICS was required in the PROC CAPABILITY statement since the INSET statement only enhances output created on high resolution graphics devices.

```
data wire;
   label strength='Torsion Strength in lb/in';
   input strength @@;
   datalines;
25 25 36 31 26 36 29 37 37 20
34 27 21 35 30 41 33 21 26 26
19 25 14 32 30 29 31 26 22 24
34 33 28 26 43 30 40 32 32 31
25 26 27 34 33 27 33 29 30 31
;
run;
```

A histogram is used to examine the data distribution. For a more complete report, the sample size, minimum value, maximum value, mean, and standard deviation are displayed on the histogram. The following statements illustrate how to inset these statistics:

```
title 'Torsion Strength of Copper Wire';
proc capability data=wire noprint;
   spec lsl=22 llsl=2 usl=38 lusl=20;
   histogram strength;
   inset n min max mean std;
run;
```

The resulting histogram is displayed in Figure 12.1. The INSET statement immediately follows the plot statement that creates the graphical display (in this case, the HISTOGRAM statement). Specify the keywords for inset statistics (such as N, MIN, MAX, MEAN, and STD) immediately after the word INSET. The inset statistics appear in the order in which you specify the keywords.

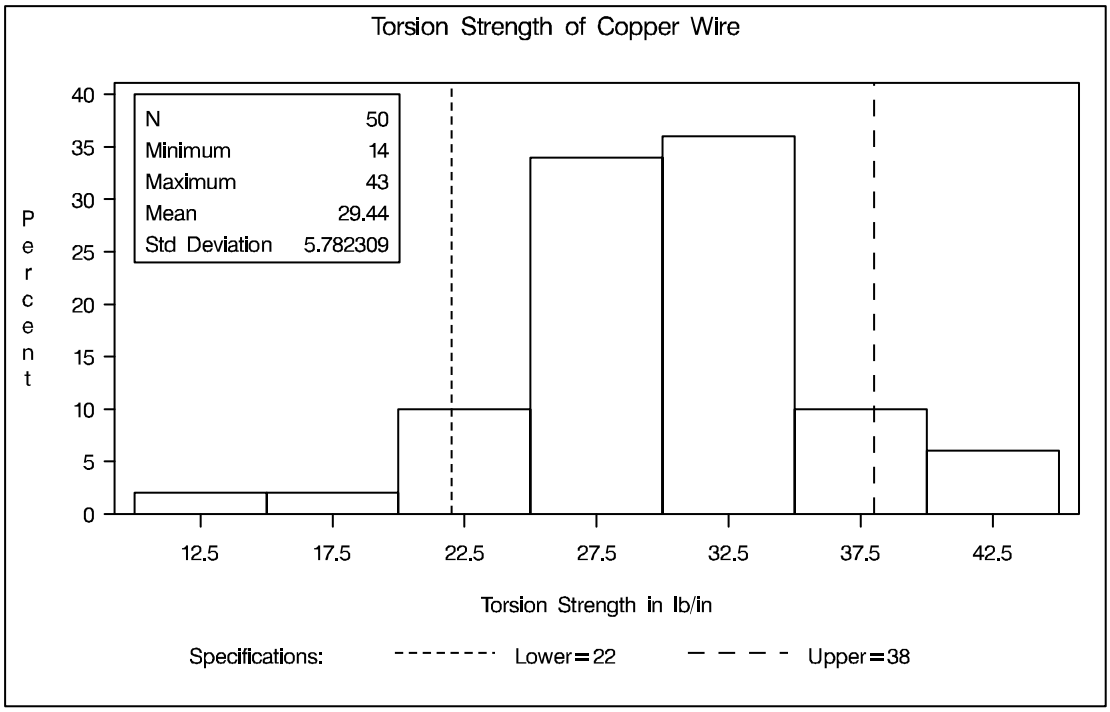


Figure 12.1. A Histogram with an Inset

A complete list of keywords that you can use with the INSET statement is provided in "Summary of INSET Keywords" on page 362. Note that the set of keywords available for a particular display depends on both the plot statement that precedes the INSET statement and the options that you specify in the plot statement.

The following examples illustrate options commonly used for enhancing the appearance of an inset.

Formatting Values and Customizing Labels

See CAPINS1 in the SAS/QC Sample Library

By default, each inset statistic is identified with an appropriate label, and each numeric value is printed using an appropriate format. However, you may want to provide your own labels and formats. For example, in Figure 12.1 the default format for the standard deviation prints an excessive number of decimal places. The following statements correct this problem, as well as customizing some of the labels displayed in the inset:

```
title 'Torsion Strength of Copper Wire';
proc capability data=wire noprint;
   spec lsl=22 llsl=2 usl=38 lusl=20;
   histogram strength;
   inset n='Sample Size' min max mean std='Std Dev' (5.2);
run;
```

The resulting histogram is displayed in Figure 12.2. You can provide your own label by specifying the keyword for that statistic followed by an equal sign (=) and the label in quotes. The label can have up to 24 characters.

The format 5.2 specified in parentheses after the keyword STD displays the standard deviation with a field width of five and two decimal places. In general, you can specify any numeric SAS format in parentheses after an inset keyword. You can also specify a format to be used for all the statistics in the INSET statement with the FORMAT= option (see the next example, "Adding a Header and Positioning the Inset"). For more information about SAS formats, refer to *SAS Language Reference: Dictionary*.

Note that if you specify both a label and a format for a statistic, the label must appear before the format, as with the keyword STD in the previous statements.

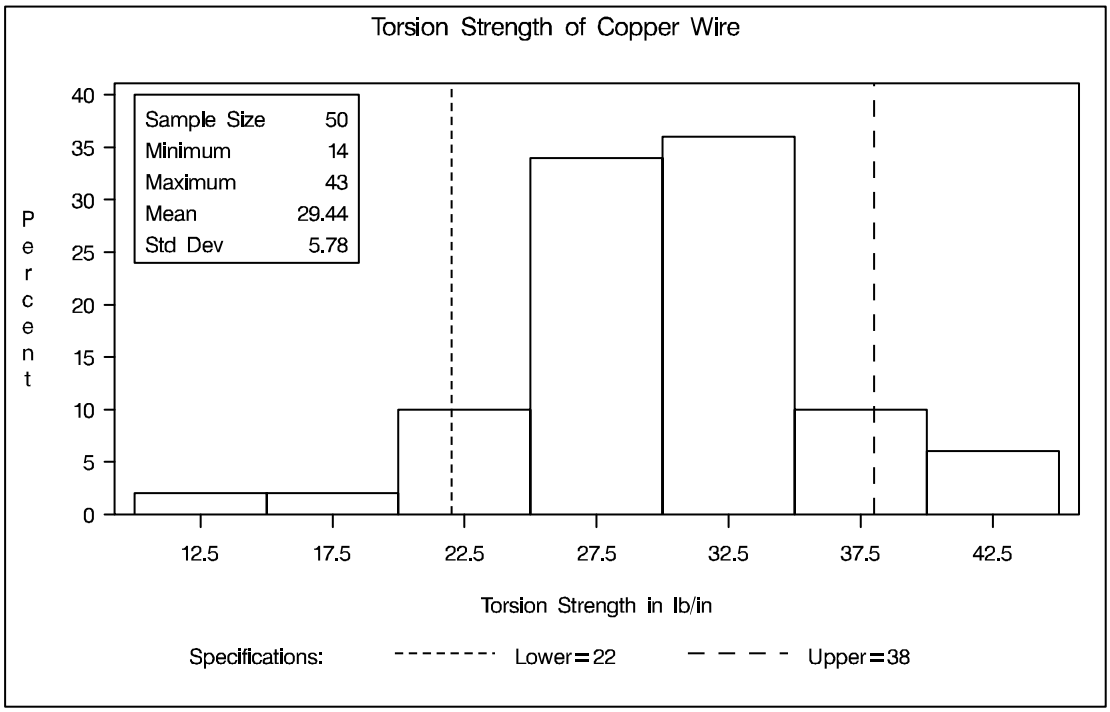


Figure 12.2. Formatting Values and Customizing Labels in an Inset

Adding a Header and Positioning the Inset

See CAPINS1 in the SAS/QC Sample Library

In the previous examples, the inset is displayed in the upper left corner of the plot, the default position for insets added to histograms. You can control the inset position with the POSITION= option. In addition, you can display a header at the top of the inset with the HEADER= option. The following statements create the chart shown in Figure 12.3:

```
title 'Torsion Strength of Copper Wire';
proc capability data=wire noprint;
   spec lsl=22 llsl=2 usl=38 lusl=20;
   histogram strength;
   inset n='Sample Size' min max range mode sum mean
      std='Standard Deviation' var stdmean skewness
      kurtosis / format = 6.1
                 pos    = rm
                 header = 'Data Summary';
run;
```

The header (in this case, *Data Summary*) can be up to 40 characters. Note that a long list of inset statistics is requested. Consequently, POSITION=RM is specified to position the inset in the right margin. For more information about positioning, see "Details" on page 370. Also note that the FORMAT= option is used to format all inset statistics. The *options*, such as HEADER=, POSITION=, and FORMAT=,

are specified after the slash (/) in the INSET statement. For more details on INSET statement options, see "Dictionary of Options" on page 368.

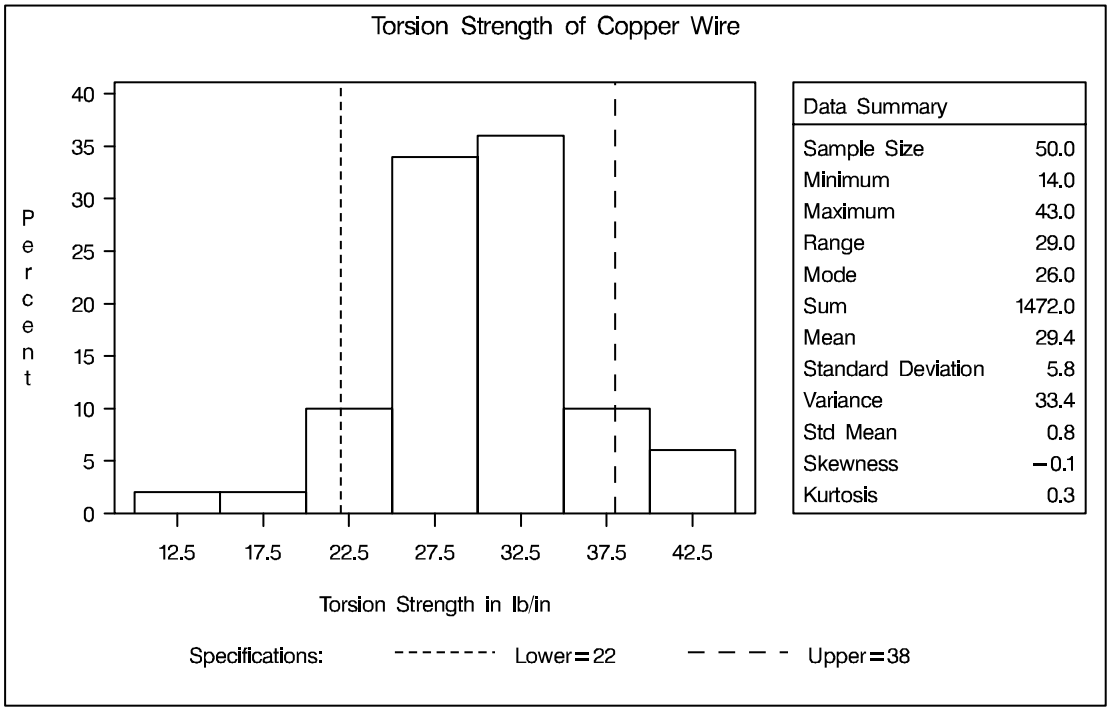


Figure 12.3. Adding a Header and Repositioning the Inset

Syntax

The syntax for the INSET statement is as follows:

INSET *keyword-list* < **/** *options* >**;**

You can use any number of INSET statements in the CAPABILITY procedure. Each INSET statement produces an inset and must follow one of the plot statements CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PPPLOT, PROBPLOT, or QQPLOT. The inset appears in all displays produced by the plot statement that immediately precedes it. The statistics are displayed in the order in which they are specified. For example, the following statements produce a cumulative distribution plot with two insets and a histogram with one inset:

```
proc capability data=wire;
   cdfplot strength;
      inset mean std min max n;
      inset p1 p5 p10;
   histogram strength;
      inset var skewness kurtosis;
run;
```

The statistics displayed in an inset are computed for a specific process variable using observations for the current BY group. For example, in the following statements, there are two process variables (STRENGTH and DIAMETER) and a BY variable (BATCH). If there are three different batches (levels of BATCH), then a total of six histograms are produced. The statistics in each inset are computed for a particular variable and batch. The labels in the inset are the same for each histogram.

```
proc capability data=wire2;
   by batch;
   histogram strength diameter / normal;
   inset mean std min max normal(mu sigma);
run;
```

The components of the INSET statement are described as follows.

keyword-list
can include any of the *keywords* listed in "Summary of INSET Keywords" on page 362. Some *keywords* allow *secondary keywords* to be specified in parentheses immediately after the *primary keyword*. Also, some inset statistics are available only if you request plot statements and options for which those statistics are calculated. For example, consider the following statements:

```
proc capability data=wire;
   histogram strength / normal;
   inset mean std normal(ad adpval);
run;
```

The *keywords* MEAN and STD display the sample mean and standard deviation of STRENGTH. The *primary keyword* NORMAL with the *secondary keywords* AD and ADPVAL display the Anderson-Darling goodness-of-fit test statistic and p-value in the inset as well. The statistics specified with the NORMAL keyword are available only because a normal distribution has been fit to the data using the NORMAL option in the HISTOGRAM statement. See the "Summary of INSET Keywords" section, which follows, for a list of available *keywords*.

Typically, you specify *keywords*, to display statistics computed by the CAPABILITY procedure. However, you can also specify the *keyword* DATA= followed by the name of a SAS data set to display customized statistics. This data set must contain two variables:

- a character variable named _LABEL_ whose values provide labels for inset entries.
- a variable named _VALUE_, which can be either character or numeric, and whose values provide values for inset entries.

The label and value from each observation in the DATA= data set occupy one line in the inset. The position of the DATA= keyword in the keyword list determines the position of its lines in the inset.

By default, inset statistics are identified with appropriate labels, and numeric values are printed using appropriate formats. However, you can provide customized labels and formats. You provide the customized label by specifying the *keyword* for that statistic followed by an equal sign (=) and the label in quotes. Labels can have up to 24 characters. You provide the numeric format in parentheses after the *keyword*. Note that if you specify both a label and a format for a statistic, the label must appear before the format. For an example, see "Formatting Values and Customizing Labels" on page 357.

options

appear after the slash (/) and control the appearance of the inset. For example, the following INSET statement uses two appearance *options* (POSITION= and CTEXT=):

```
inset mean std min max / position=ne ctext=yellow;
```

The POSITION= option determines the location of the inset, and the CTEXT= option specifies the color of the text of the inset.

See "Summary of Options" on page 367 for a list of all available *options*, and "Dictionary of Options" on page 368 for detailed descriptions. Note the difference between *keywords* and *options*; *keywords* specify the information to be displayed in an inset, whereas *options* control the appearance of the inset.

Summary of INSET Keywords

Summary Statistics and Process Capability Indices

Table 12.1. Summary Statistics

N	sample size
SUMWGT	sum of the weights
MEAN	sample mean
SUM	sum of the observations
STD	standard deviation
VAR	variance
SKEWNESS	skewness
KURTOSIS	kurtosis
MAX	largest value
MIN	smallest value
NOBS	number of observations
RANGE	range
MODE	most frequent value
NMISS	number of missing values
USS	uncorrected sum of squares
CSS	corrected sum of squares
CV	coefficient of variation
STDMEAN	standard error of the mean
NEXCL	number of observations excluded by MAXSIGMAS= option (COMPHISTOGRAM statement only)
DATA=	arbitrary values from *SAS-data-set*

Table 12.2. Percentile Statistics

P1	1^{st} percentile
P5	5^{th} percentile
P10	10^{th} percentile
Q1	lower quartile (25^{th} percentile)
MEDIAN	median (50^{th} percentile)
Q3	upper quartile (75^{th} percentile)
P90	90^{th} percentile
P95	95^{th} percentile
P99	99^{th} percentile
QRANGE	interquartile range (Q3 - Q1)

Table 12.3. Test of Normality

NORMALTEST	test statistic for normality
PNORMAL	probability value for the normality test

Table 12.4. Signed Rank Test

SIGNRANK	signed rank statistic
PROBS	probability value for the signed rank test

Table 12.5. Capability Indices and Confidence Limits

CP	capability index C_p
CPLCL	lower confidence limit for C_p
CPUCL	upper confidence limit for C_p
CPK	capability index C_{pk}
CPKLCL	lower confidence limit for C_{pk}
CPKUCL	upper confidence limit for C_{pk}
CPL	capability index *CPL*
CPM	capability index C_{pm}
CPMLCL	lower confidence limit for C_{pm}
CPMUCL	upper confidence interval for C_{pm}
CPU	capability index *CPU*
K	capability index K

Table 12.6. Specification Limits and Related Information

LSL	lower specification limit
USL	upper specification limit
TARGET	target value
PCTGTR	percent of nonmissing observations that exceed the upper specification limit
PCTLSS	percent of nonmissing observations that are less than the lower specification limit
PCTBET	percent of nonmissing observations between the upper and lower specification limits (inclusive)

Table 12.7. Student's t-Test

T	statistic for Student's t-test
PROBT	probability value for Student's t-test

Statistics Available with Parametric Density Estimates

You can request parametric density estimates with all plot statements in the CAPABILITY procedure (CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PPPLOT, PROBPLOT, and QQPLOT). You can display parameters and statistics associated with these estimates in an inset by specifying a distribution keyword followed by secondary keywords in parentheses. For example, the following statements create a histogram for STRENGTH with a fitted exponential density curve:

```
proc capability data=wire;
   histogram strength / exp;
   inset exp(sigma theta);
run;
```

The secondary keywords SIGMA and THETA for the EXP distribution keyword request an inset displaying the values of the exponential scale parameter σ and threshold parameter θ. You must request the distribution option in the plot statement to display the corresponding distribution statistics in an inset. Specifying a distribution keyword

with no secondary keywords produces an inset displaying the full set of parameters for that distribution. See Output 12.1.1 on page 375 for an example of an inset with statistics from a fitted normal curve.

The following table describes the available distribution keywords. Note that some keywords are not available with all plot statements.

Table 12.8. Density Estimation Primary Keywords

Keyword	Distribution	Plot Statement Availability
BETA	beta	all except COMPHISTOGRAM
EXPONENTIAL	exponential	all except COMPHISTOGRAM
GAMMA	gamma	all except COMPHISTOGRAM
LOGNORMAL	lognormal	all except COMPHISTOGRAM
NORMAL	normal	all plot statements
SB	Johnson S_B	all except COMPHISTOGRAM
SU	Johnson S_U	all except COMPHISTOGRAM
WEIBULL	Weibull	all except COMPHISTOGRAM
WEIBULL2	2-parameter Weibull	PROBPLOT and QQPLOT

Table 12.9 through Table 12.17 list the secondary keywords available with each distribution keyword listed in Table 12.8. In many cases, aliases can be used (for example, ALPHA in place of SHAPE1).

Table 12.9. Secondary Keywords Available with the BETA Keyword

Secondary Keyword	Alias	Description
ALPHA	SHAPE1	first shape parameter α
BETA	SHAPE2	second shape parameter β
SIGMA	SCALE	scale parameter σ
THETA	THRESHOLD	lower threshold parameter θ
MEAN		mean of the fitted distribution
STD		standard deviation of the fitted distribution

Table 12.10. Secondary Keywords Available with the EXP Keyword

Secondary Keyword	Alias	Description
SIGMA	SCALE	scale parameter σ
THETA	THRESHOLD	threshold parameter θ
MEAN		mean of the fitted distribution
STD		standard deviation of the fitted distribution

Table 12.11. Secondary Keywords Available with the GAMMA Keyword

Secondary Keyword	Alias	Description
ALPHA	SHAPE	shape parameter α
SIGMA	SCALE	scale parameter σ
THETA	THRESHOLD	threshold parameter θ
MEAN		mean of the fitted distribution
STD		standard deviation of the fitted distribution

Table 12.12. Secondary Keywords Available with the LOGNORMAL Keyword

Secondary Keyword	Alias	Description
SIGMA	SHAPE	shape parameter σ
THETA	THRESHOLD	threshold parameter θ
ZETA	SCALE	scale parameter ζ
MEAN		mean of the fitted distribution
STD		standard deviation of the fitted distribution

Table 12.13. Secondary Keywords Available with the NORMAL Keyword

Secondary Keyword	Alias	Description
MU	MEAN	mean parameter μ
SIGMA	STD	scale parameter σ

Table 12.14. Secondary Keywords Available with the SB Keyword

Secondary Keyword	Alias	Description
DELTA		shape parameter δ
GAMMA		shape parameter γ
SIGMA	SHAPE	scale parameter σ
THETA	THRESHOLD	threshold parameter θ
MEAN		mean of the fitted distribution
STD		standard deviation of the fitted distribution

Table 12.15. Secondary Keywords Available with the SU Keyword

Secondary Keyword	Alias	Description
DELTA		shape parameter δ
GAMMA		shape parameter γ
SIGMA	SHAPE	scale parameter σ
THETA		location parameter θ
MEAN		mean of the fitted distribution
STD		standard deviation of the fitted distribution

Table 12.16. Secondary Keywords Available with the WEIBULL Keyword

Secondary Keyword	Alias	Description
C	SHAPE	shape parameter c
SIGMA	SCALE	scale parameter σ
THETA	THRESHOLD	threshold parameter θ
MEAN		mean of the fitted distribution
STD		standard deviation of the fitted distribution

Table 12.17. Secondary Keywords Available with the WEIBULL2 Keyword

Secondary Keyword	Alias	Description
C	SHAPE	shape parameter c
SIGMA	SCALE	scale parameter σ
THETA	THRESHOLD	known lower threshold θ_0
MEAN		mean of the fitted distribution
STD		standard deviation of the fitted distribution

The secondary keywords listed in Table 12.18 can be used with any distribution keyword but *only* with the HISTOGRAM and COMPHISTOGRAM plot statements.

Table 12.18. Statistics Computed from Any Parametric Density Estimate

Secondary Keyword	Description
CP	capability index C_p
CPK	capability index C_{pk}
CPL	capability index *CPL*
CPM	capability index C_{pm}
CPU	capability index *CPU*
ESTPCTLSS	estimated percentage less than the lower specification limit
ESTPCTGTR	estimated percentage greater than the upper specification limit
K	capability index K

The secondary keywords listed in Table 12.19 can be used with any distribution keyword but *only* with the HISTOGRAM plot statement (see Example 12.1 on page 374).

Table 12.19. Goodness-of-Fit Statistics for Fitted Curves

Secondary Keyword	Description
CHISQ	chi-square statistic
DF	degrees of freedom for the chi-square test
PCHISQ	probability value for the chi-square test
AD	Anderson-Darling EDF test statistic
ADPVAL	Anderson-Darling EDF test p-value
CVM	Cramér-von Mises EDF test statistic
CVMPVAL	Cramér-von Mises EDF test p-value
KSD	Kolmogorov-Smirnov EDF test statistic
KSDPVAL	Kolmogorov-Smirnov EDF test p-value

Table 12.20 lists primary keywords available only with the HISTOGRAM and COMPHISTOGRAM plot statements. These keywords display fill areas on a histogram. If you fit a parametric density on a histogram and request that the area under the curve be filled, these keywords display the percentage of the distribution area that lies below the lower specification limit, between the specification limits, or above the upper specification limit. If you do not fill the area beneath a parametric density estimate, these keywords display the observed proportion of observations (that is, the area in the bars of the histogram).

You should use these options with the FILL, CFILL=, and PFILL= options in the HISTOGRAM and COMPHISTOGRAM statements and with the CLEFT=, CRIGHT=, PLEFT=, and PRIGHT= options in the SPEC statements. See Output 12.2.1 on page 376 for an example.

Table 12.20. Curve Area Keywords

Keyword	Alias	Description
BETWEENPCT	BETPCT	area between the specification limits
LSLPCT		area below the lower specification limit
USLPCT		area above the upper specification limit

Statistics Available with Nonparametric Kernel Density Estimates

You can request nonparametric kernel density estimates with the HISTOGRAM and COMPHISTOGRAM plot statements. You can display statistics associated with

these estimates by specifying a kernel density keyword followed by secondary keywords in parentheses. For example, the following statements create a histogram for STRENGTH with a fitted kernel density estimate:

```
proc capability data=wire;
   histogram strength / kernel;
   inset kernel(c amise);
run;
```

The secondary keywords C and AMISE for the KERNEL keyword display the values of the standardized bandwidth c and the approximate mean integrated square error.

Note that you can specify up to five kernel density estimates on a single histogram. If you specify multiple kernel density estimates, you can request inset statistics for all of the estimates with the KERNEL keyword, or you can display inset statistics for individual curves with KERNELn keywords, as in the following example:

```
proc capability data=wire;
   histogram strength / kernel(c = 1 2 3);
   inset kernel2(c) kernel3(c);
run;
```

Three kernel density estimates are displayed on the histogram, but the inset displays the value of c only for the second and third estimates.

Table 12.21 lists the kernel density keywords. Table 12.22 lists the available secondary keywords.

Table 12.21. Kernel Density Estimate Primary Keywords

Keyword	Description
KERNEL	displays statistics for all kernel estimates
KERNELn	displays statistics for only the n^{th} kernel density estimate $n = 1, 2, 3, 4,$ or 5

Table 12.22. Secondary Keywords Available with the KERNEL Keyword

Secondary Keyword	Description
TYPE	kernel type: normal, quadratic, or triangular
BANDWIDTH	bandwidth λ for the density estimate
BWIDTH	alias for BANDWIDTH
C	standardized bandwidth c for the density estimate: $c = \frac{\lambda}{Q} n^{\frac{1}{5}}$ where $n =$ sample size, $\lambda =$ bandwidth, and $Q =$ interquartile range
AMISE	approximate mean integrated square error (MISE) for the kernel density

Summary of Options

The following table lists the INSET statement options. For complete descriptions, see "Dictionary of Options," which follows this section.

Table 12.23. INSET Options

CFILL=*color* | BLANK	specifies color of inset background
CFILLH=*color*	specifies color of header background
CFRAME=*color*	specifies color of frame
CHEADER=*color*	specifies color of header text
CSHADOW=*color*	specifies color of drop shadow
CTEXT=*color*	specifies color of inset text
DATA	specifies data units for POSITION=(x, y) coordinates
FONT=*font*	specifies font of text
FORMAT=*format*	specifies format of values in inset
HEADER=*'quoted string'*	specifies header text
HEIGHT=*value*	specifies height of inset text
NOFRAME	suppresses frame around inset
POSITION=*position*	specifies position of inset
REFPOINT=BR|BL|TR|TL	specifies reference point of inset positioned with POSITION=(x, y) coordinates

Dictionary of Options

The following entries provide detailed descriptions of options for the INSET statement. Terms used in this section are illustrated in Figure 12.4.

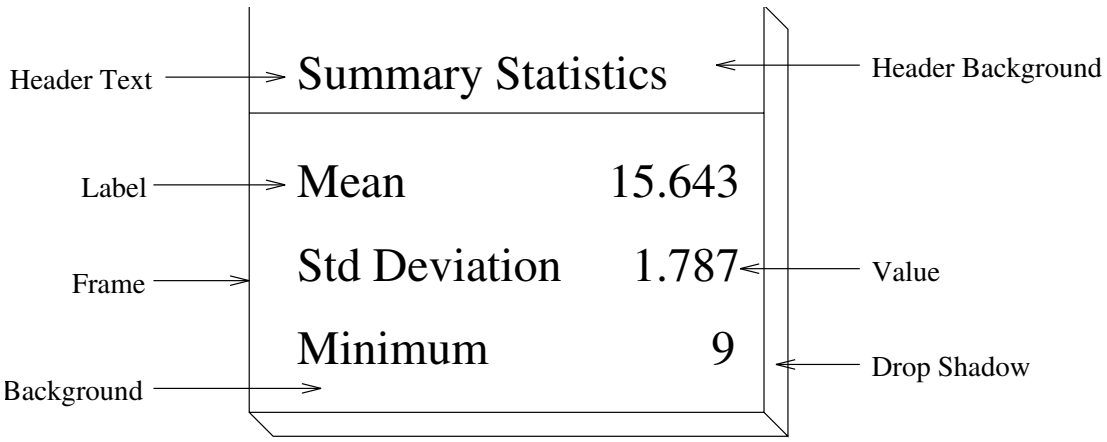


Figure 12.4. The Inset

CFILL=*color* | BLANK

specifies the color of the background (including the header background if you do not specify the CFILLH= option). See Output 12.1.1 on page 375 for an example.

If you do not specify the CFILL= option, then by default, the background is empty. This means that items that overlap the inset (such as curves, histogram bars, or specification limits) show through the inset. If you specify any value for the CFILL= option,

then overlapping items no longer show through the inset. Specify CFILL=BLANK to leave the background uncolored and also to prevent items from showing through the inset.

CFILLH=*color*
specifies the color of the header background. By default, if you do not specify a CFILLH= color, the CFILL= color is used.

CFRAME=*color*
specifies the color of the frame. By default, the frame is the same color as the axis of the plot.

CHEADER=*color*
specifies the color of the header text. By default, if you do not specify a CHEADER= color, the CTEXT= color is used.

CSHADOW=*color*
CS=*color*
specifies the color of the drop shadow. See Output 12.2.1 on page 376 for an example. By default, if you do not specify the CSHADOW= option, a drop shadow is not displayed.

CTEXT=*color*
CT=*color*
specifies the color of the text. By default, the inset text color is the same as the other text on the plot.

DATA
specifies that data coordinates are to be used in positioning the inset with the POSITION= option. The DATA option is available only when you specify POSITION= (x, y), and it must be placed immediately after the coordinates (x, y). For details, see the entry for the POSITION= option or "Positioning the Inset Using Coordinates" on page 372. See Figure 12.7 on page 373 for an example.

FONT=*font*
specifies the font of the text. By default, the font is SIMPLEX if the inset is located in the interior of the plot, and the font is the same as the other text displayed on the plot if the inset is located in the exterior of the plot.

FORMAT=*format*
specifies a format for all the values displayed in an inset. If you specify a format for a particular statistic, then this format overrides the format you specified with the FORMAT= option. See Figure 12.3 on page 359 or Output 12.1.1 on page 375 for an example.

HEADER= *'string'*
specifies the header text. The *string* cannot exceed 40 characters. If you do not specify the HEADER= option, no header line appears in the inset. If all the keywords listed in the INSET statement are secondary keywords corresponding to a fitted curve on a histogram, a default header is displayed that indicates the distribution and identifies the curve. See Figure 12.3 on page 359 for an example of a specified header and

Output 12.1.1 on page 375 for an example of the default header for a fitted normal curve.

HEIGHT=*value*
specifies the height of the text.

NOFRAME
suppresses the frame drawn around the text.

POSITION=*position*
POS=*position*
determines the position of the inset. The *position* can be a compass point keyword, a margin keyword, or a pair of coordinates (x, y). You can specify coordinates in axis percent units or axis data units. For more information, see "Details" on page 370. By default, POSITION=NW, which positions the inset in the upper left (northwest) corner of the display.

REFPOINT=BR | BL | TR | TL
RP=BR | BL | TR | TL
specifies the reference point for an inset that is positioned by a pair of coordinates with the POSITION= option. Use the REFPOINT= option with POSITION= coordinates. The REFPOINT= option specifies which corner of the inset frame you want positioned at coordinates (x, y). The keywords BL, BR, TL, and TR represent bottom left, bottom right, top left, and top right, respectively. See Figure 12.8 on page 374 for an example. The default is REFPOINT=BL.

If you specify the position of the inset as a compass point or margin keyword, the REFPOINT= option is ignored. For more information, see "Positioning the Inset Using Coordinates" on page 372.

Details

This section provides details on three different methods of positioning the inset using the POSITION= option. With the POSITION= option, you can specify

- compass points
- keywords for margin positions
- coordinates in data units or percent axis units

Positioning the Inset Using Compass Points

See CAPINS2 in the SAS/QC Sample Library

You can specify the eight compass points N, NE, E, SE, S, SW, W, and NW as keywords for the POSITION= option. The following statements create the display in Figure 12.5, which demonstrates all eight compass positions. The default is NW.

```
title 'Torsion Strength of Copper Wire';
proc capability data=wire;
   histogram strength / cfill=gray;
   inset n     / cfill=blank header='Position = NW' pos=nw;
   inset mean  / cfill=blank header='Position = N ' pos=n ;
   inset sum   / cfill=blank header='Position = NE' pos=ne;
   inset max   / cfill=blank header='Position = E ' pos=e ;
   inset min   / cfill=blank header='Position = SE' pos=se;
   inset nobs  / cfill=blank header='Position = S ' pos=s ;
   inset range / cfill=blank header='Position = SW' pos=sw;
   inset mode  / cfill=blank header='Position = W ' pos=w ;
run;
```

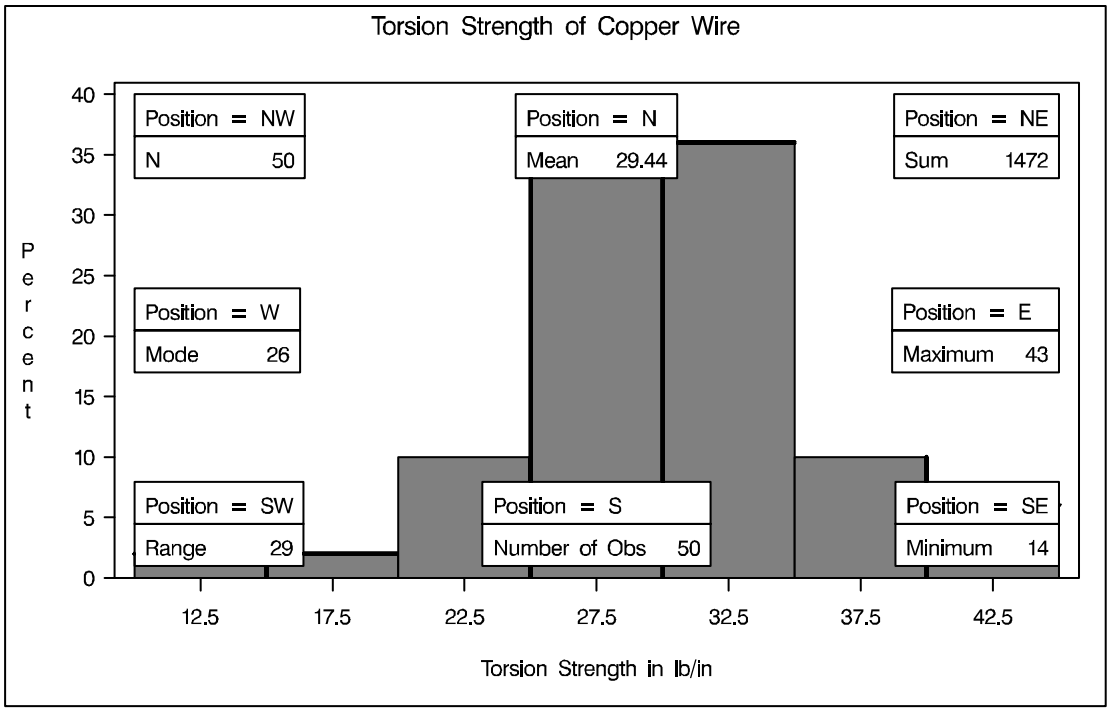


Figure 12.5. Insets Positioned Using Compass Points

Positioning the Inset in the Margins

You can also position the inset in one of the four margins surrounding the plot area using the margin keywords LM, RM, TM, or BM, as illustrated in Figure 12.6.

For an example of an inset placed in the right margin, see Figure 12.3 on page 359. Margin positions are recommended if a large number of statistics are listed in the INSET statement. If you attempt to display a lengthy inset in the interior of the plot, it is likely that the inset will collide with the data display.

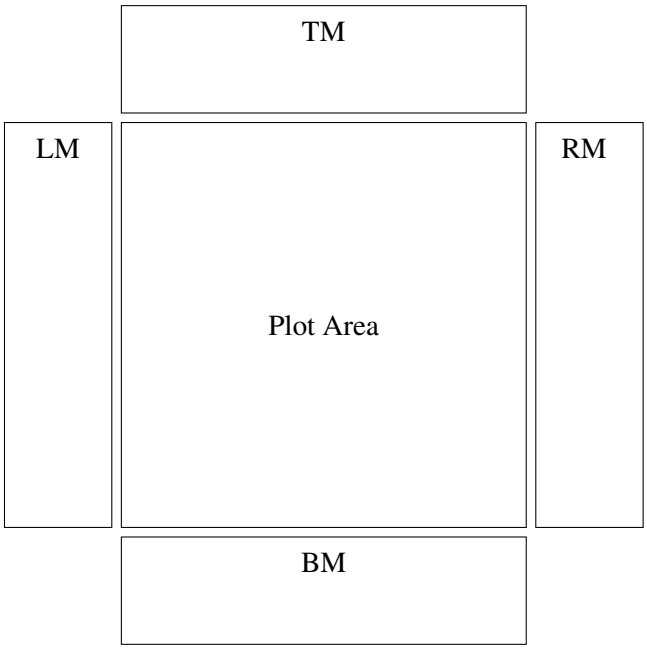


Figure 12.6. Positioning Insets in the Margins

Positioning the Inset Using Coordinates

You can also specify the position of the inset with coordinates: POSITION= (x, y). The coordinates can be given in axis percent units (the default) or in axis data units.

Data Unit Coordinates

See CAPINS2 in the SAS/QC Sample Library

If you specify the DATA option immediately following the coordinates, the inset is positioned using axis data units. For example, the following statements place the bottom left corner of the inset at 12.5 on the horizontal axis and 10 on the vertical axis:

```
title 'Torsion Strength of Copper Wire';
proc capability data=wire;
   histogram strength / cfill=gray;
   inset n / header   = 'Position=(12.5,10)'
             position = (12.5,10) data;
run;
```

The histogram is displayed in Figure 12.7. By default, the specified coordinates determine the position of the bottom left corner of the inset. You can change this reference point with the REFPOINT= option, as in the next example.

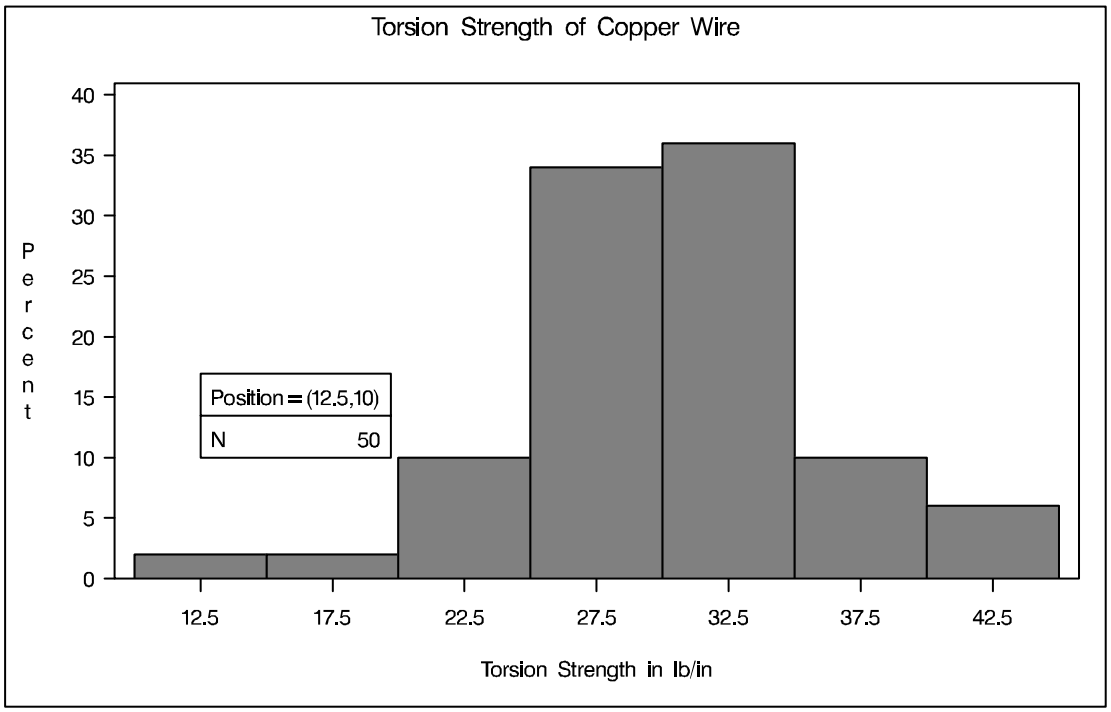


Figure 12.7. Inset Positioned Using Data Unit Coordinates

Axis Percent Unit Coordinates

If you do not use the DATA option, the inset is positioned using axis percent units. The coordinates of the bottom left corner of the display are $(0, 0)$, while the upper right corner is $(100, 100)$. For example, the following statements create a histogram with two insets, both positioned using coordinates in axis percent units:

See CAPINS2 in the SAS/QC Sample Library

```
title 'Torsion Strength of Copper Wire';
proc capability data=wire;
   histogram strength / cfill=gray;
   inset min / position = (5,25)
               header   = 'Position=(5,25)'
               refpoint = tl;
   inset max / position = (95,95)
               header   = 'Position=(95,95)'
               refpoint = tr;
run;
```

The display is shown in Figure 12.8. Notice that the REFPOINT= option is used to determine which corner of the inset is to be placed at the coordinates specified with the POSITION= option. The first inset has REFPOINT=TL, so the top left corner of the inset is positioned 5% of the way across the horizontal axis and 25% of the way up the vertical axis. The second inset has REFPOINT=TR, so the top right corner of the inset is positioned 95% of the way across the horizontal axis and 95% of the way

up the vertical axis. Note also that coordinates in axis percent units must be *between* 0 and 100.

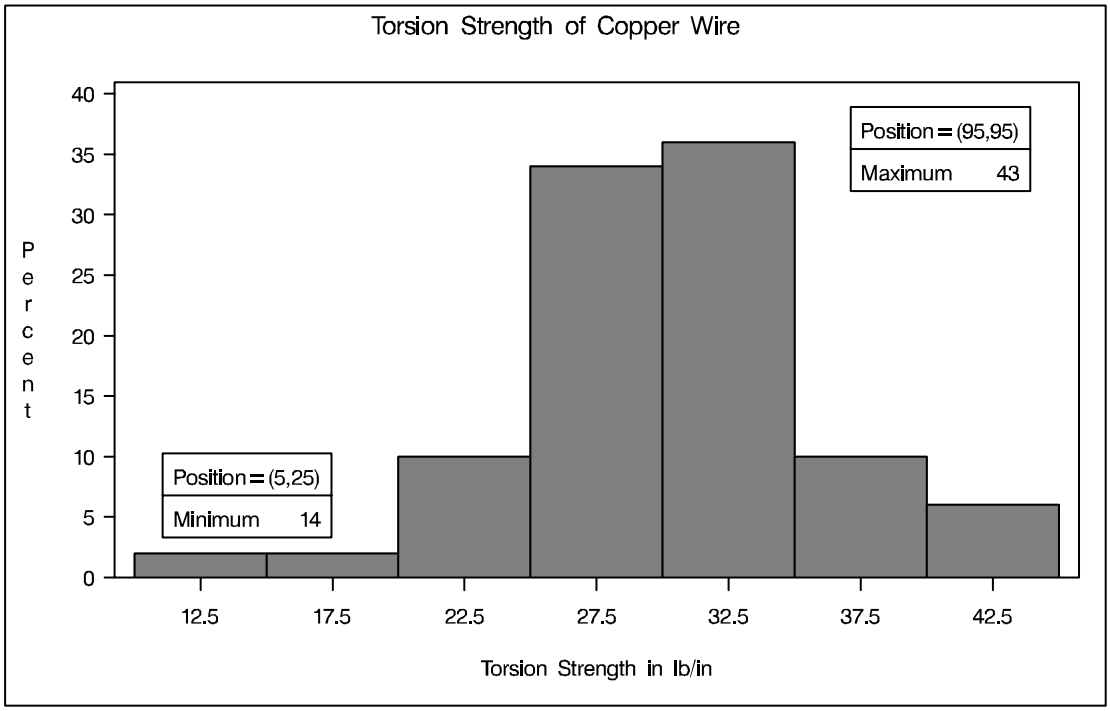


Figure 12.8. Inset Positioned Using Axis Percent Unit Coordinates

Examples

This section provides advanced examples using the INSET statement.

Example 12.1. Inset for Goodness-of-Fit Statistics

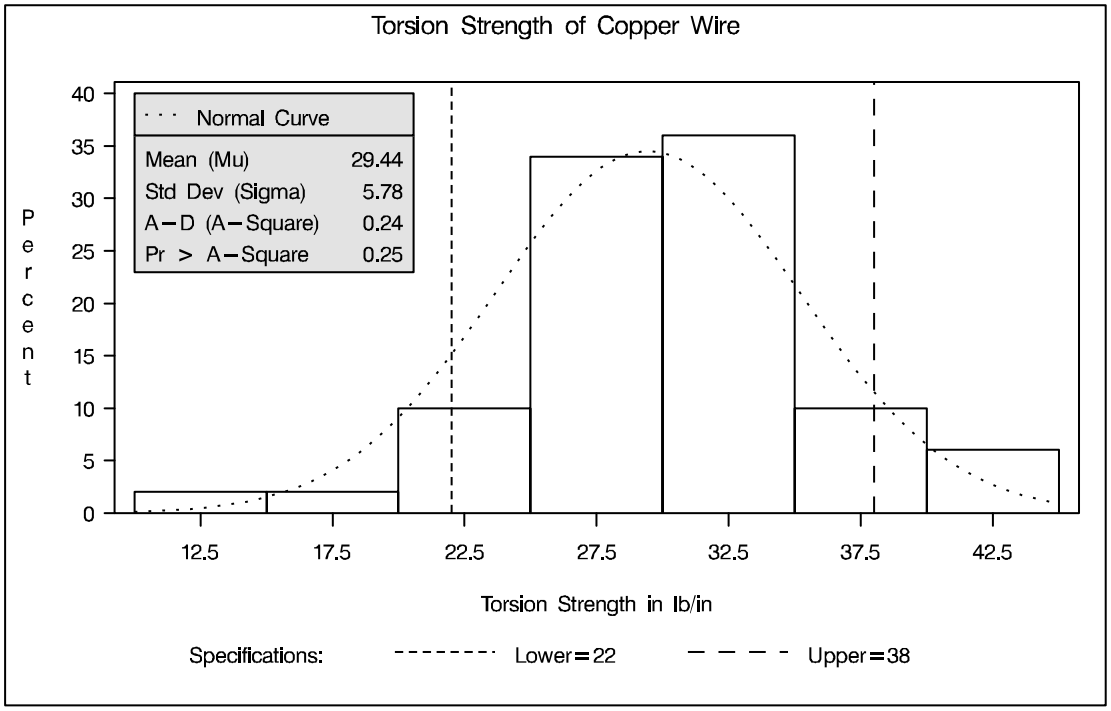
See CAPINS3 in the SAS/QC Sample Library

This example fits a normal curve to the torsion strength data used in the "Getting Started" section on page 355. The following statements fit a normal curve and request an inset summarizing the fitted curve with the mean, the standard deviation, and the Anderson-Darling goodness-of-fit test:

```
title 'Torsion Strength of Copper Wire';
proc capability data=wire noprint;
   spec lsl=22 llsl=2 usl=38 lusl=20;
   histogram strength /
                        normal(color=black noprint l=34)
                        nocurvelegend;
   inset normal(mu sigma ad adpval) / cfill  = yellow
                                      format = 7.2;
run;
```

The resulting histogram is displayed in Output 12.1.1. The NOCURVELEGEND option in the HISTOGRAM statement suppresses the default legend for curve parameters.

Output 12.1.1. Inset Table with Normal Curve Information



Example 12.2. Inset for Areas Under a Fitted Curve

See CAPINS4 in the SAS/QC Sample Library

You can use the INSET keywords LSLPCT, USLPCT, and BETWEENPCT to inset legends for areas under histogram bars or fitted curves. The following statements create a histogram with an inset legend for the shaded area under the fitted normal curve to the left of the lower specification limit:

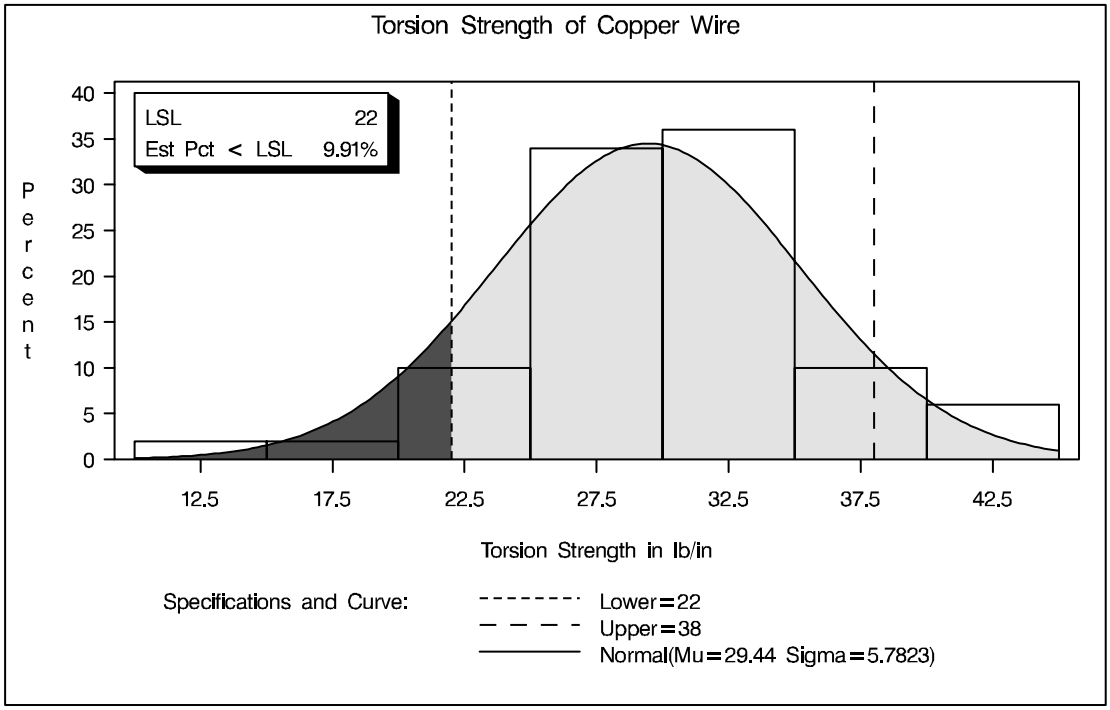
```
title 'Torsion Strength of Copper Wire';
proc capability data=wire noprint;
   spec lsl=22 llsl=2 cleft=red
        usl=38 lusl=20;
   histogram strength /
                          cfill  = yellow
                          normal(color=black noprint fill);
   inset  lsl='LSL' lslpct / cshadow=black;
run;
```

The histogram is displayed in Output 12.2.1. The LSLPCT keyword in the INSET statement requests a legend for the area under the curve to the left of the lower specification limit. The CLEFT= option is used to fill the area under the normal curve

to the left of the line, and the CFILL= color is used to fill the remaining area. If the FILL *normal-option* were not specified, the CLEFT= and CFILL= colors would be applied to the corresponding areas under the histogram, not the normal curve, and the inset box would reflect the area under the histogram bars.

You can use the USLPCT keyword in the INSET statement to request a legend for the area to the right of an upper specification limit, and you can use the BETWEENPCT keyword to request a legend for the area between the lower and upper limits. By default, the legend requested with each of the keywords LSLPCT, USLPCT, and BETWEENPCT displays a rectangle that matches the color of the corresponding area. You can substitute a customized label for each rectangle by specifying the keyword followed by an equal sign (=) and the label in quotes.

Output 12.2.1. Displaying Areas Under the Normal Curve



Chapter 13
INTERVALS Statement

Chapter Contents

OVERVIEW 379

GETTING STARTED 379
- Computing Statistical Intervals 380
- Computing One-Sided Lower Prediction Limits 382

SYNTAX 383
- Summary of Options 384
- Dictionary of Options 384

DETAILS 386
- Methods for Computing Statistical Intervals 386
- OUTINTERVALS= Data Set 389
- ODS Tables 390

Chapter 13
INTERVALS Statement

Overview

The INTERVALS statement tabulates various statistical intervals for selected process variables. The types of intervals you can request include

- approximate simultaneous prediction intervals for future observations
- prediction intervals for the mean of future observations
- approximate statistical tolerance intervals that contain at least a specified proportion of the population
- confidence intervals for the population mean
- prediction intervals for the standard deviation of future observations
- confidence intervals for the population standard deviation

These intervals are computed assuming the data are sampled from a normal population. See Hahn and Meeker (1991) for a detailed discussion of these intervals.

You can use options in the INTERVALS statement to

- specify which intervals to compute
- provide probability or confidence levels for intervals
- suppress printing of output tables
- create an output data set containing interval information
- specify interval type (one-sided lower, one-sided upper, or two-sided)

Getting Started

This section introduces the INTERVALS statement with simple examples that illustrate commonly used options. Complete syntax for the INTERVALS statement is presented in the "Syntax" section on page 383.

Computing Statistical Intervals

See CAPINT1 in the SAS/QC Sample Library

The following statements create the data set CANS, which contains measurements (in ounces) of the fluid weights of 100 drink cans. The filling process is assumed to be in statistical control.

```
data cans;
   label weight = "Fluid Weight (ounces)";
   input weight @@;
datalines;
12.07  12.02  12.00  12.01  11.98  11.96  12.04  12.05  12.01  11.97
12.03  12.03  12.00  12.04  11.96  12.02  12.06  12.00  12.02  11.91
12.05  11.98  11.91  12.01  12.06  12.02  12.05  11.90  12.07  11.98
12.02  12.11  12.00  11.99  11.95  11.98  12.05  12.00  12.10  12.04
12.06  12.04  11.99  12.06  11.99  12.07  11.96  11.97  12.00  11.97
12.09  11.99  11.95  11.99  11.99  11.96  11.94  12.03  12.09  12.03
11.99  12.00  12.05  12.04  12.05  12.01  11.97  11.93  12.00  11.97
12.13  12.07  12.00  11.96  11.99  11.97  12.05  11.94  11.99  12.02
11.95  11.99  11.91  12.06  12.03  12.06  12.05  12.04  12.03  11.98
12.05  12.05  12.11  11.96  12.00  11.96  11.96  12.00  12.01  11.98
;
run;
```

Note that this data set is introduced in "Computing Descriptive Statistics" on page 166 of Chapter 8, "PROC CAPABILITY and General Statements." The analysis in that section provides evidence that the weight measurements are normally distributed.

By default, the INTERVALS statement computes and prints the six intervals described in the entry for the METHODS= option on page 385. The following statements tabulate these intervals for the variable WEIGHT:

```
title 'Statistical Intervals for Fluid Weight';
proc capability data=cans noprint;
   intervals weight;
run;
```

The intervals are displayed in Figure 13.1 on page 381 and Figure 13.2 on page 382.

Statistical Intervals for Fluid Weight

The CAPABILITY Procedure
Two-Sided Statistical Intervals for weight Assuming Normality

Approximate Prediction Interval
Containing All of k
Future Observations

Confidence	k	Prediction Limits	
99.00%	1	11.89	12.13
99.00%	2	11.87	12.14
99.00%	3	11.87	12.15
95.00%	1	11.92	12.10
95.00%	2	11.90	12.12
95.00%	3	11.89	12.12
90.00%	1	11.93	12.09
90.00%	2	11.92	12.10
90.00%	3	11.91	12.11

Approximate Prediction Interval
Containing the Mean of
k Future Observations

Confidence	k	Prediction Limits	
99.00%	1	11.89	12.13
99.00%	2	11.92	12.10
99.00%	3	11.94	12.08
95.00%	1	11.92	12.10
95.00%	2	11.94	12.08
95.00%	3	11.95	12.06
90.00%	1	11.93	12.09
90.00%	2	11.95	12.06
90.00%	3	11.96	12.05

Approximate Tolerance Interval
Containing At Least Proportion
p of the Population

Confidence	p	Tolerance Limits	
99.00%	0.900	11.92	12.10
99.00%	0.950	11.90	12.12
99.00%	0.990	11.86	12.15
95.00%	0.900	11.92	12.10
95.00%	0.950	11.90	12.11
95.00%	0.990	11.87	12.15
90.00%	0.900	11.92	12.09
90.00%	0.950	11.91	12.11
90.00%	0.990	11.88	12.14

Figure 13.1. Statistical Intervals for WEIGHT

Statistical Intervals for Fluid Weight

Two-Sided Statistical Intervals for weight Assuming Normality

Confidence Limits Containing the Mean

Confidence	Confidence Limits	
99.00%	11.997	12.022
95.00%	12.000	12.019
90.00%	12.002	12.017

Prediction Interval Containing the Standard Deviation of k Future Observations

Confidence	k	Prediction Limits	
99.00%	2	0.0003	0.1348
99.00%	3	0.0033	0.1110
95.00%	2	0.0015	0.1069
95.00%	3	0.0075	0.0919
90.00%	2	0.0030	0.0932
90.00%	3	0.0106	0.0825

Confidence Limits Containing the Standard Deviation

Confidence	Confidence Limits	
99.00%	0.040	0.057
95.00%	0.041	0.055
90.00%	0.042	0.053

Figure 13.2. Statistical Intervals for WEIGHT(continued)

Computing One-Sided Lower Prediction Limits

See CAPINT1 in the SAS/QC Sample Library

You can specify options after the slash (/) in the INTERVALS statement to control the computation and printing of intervals. The following statements produce a table of one-sided lower prediction limits for the mean, which is displayed in Figure 13.3:

```
title 'Statistical Intervals for Fluid Weight';
proc capability data=cans noprint;
   intervals weight / methods = 1 2
                      type    = lower;
run;
```

The METHODS= option specifies which intervals to compute, and the TYPE= option requests one-sided lower limits. All the options available in the INTERVALS statement are listed in "Summary of Options" on page 384 and are described in "Dictionary of Options" on page 384.

Statistical Intervals for Fluid Weight

The CAPABILITY Procedure
One-Sided Lower Statistical Intervals for weight Assuming Normality

Approximate Prediction Limit For All of k Future Observations

Confidence	k	Lower Limit
99.00%	1	11.90
99.00%	2	11.89
99.00%	3	11.88
95.00%	1	11.93
95.00%	2	11.92
95.00%	3	11.91
90.00%	1	11.95
90.00%	2	11.93
90.00%	3	11.92

Approximate Prediction Limit For the Mean of k Future Observations

Confidence	k	Lower Limit
99.00%	1	11.90
99.00%	2	11.93
99.00%	3	11.94
95.00%	1	11.93
95.00%	2	11.95
95.00%	3	11.96
90.00%	1	11.95
90.00%	2	11.97
90.00%	3	11.97

Figure 13.3. One-Sided Lower Prediction Limits for the Mean

Syntax

The syntax for the INTERVALS statement is as follows:

INTERVALS <*variables* > < **/** *options* >**;**

You can specify INTERVAL as an alias for INTERVALS. You can use any number of INTERVALS statements in the CAPABILITY procedure. The components of the INTERVALS statement are described as follows.

variables
gives a list of variables for which to compute intervals. If you specify a VAR statement, the *variables* must also be listed in the VAR statement. Otherwise, the *variables* can be any numeric variable in the input data set. If you do not specify a list of *vari-*

ables, then by default the INTERVALS statement computes intervals for all variables in the VAR statement (or all numeric variables in the input data set if you do not use a VAR statement).

options
alter the defaults for computing and printing intervals and for creating output data sets.

Summary of Options

The following tables list the INTERVALS statement options by function. For complete descriptions, see "Dictionary of Options" on page 384.

Table 13.1. INTERVAL Statement Options

ALPHA=*value-list*	lists probability or confidence levels associated with the intervals
K=*value-list*	lists values of k for prediction intervals
METHODS=*indices*	specifies which intervals are computed
NOPRINT	suppresses the output tables
OUTINTERVALS=*SAS-data-set*	specifies an output data set containing interval information
P=*value-list*	lists values of p for tolerance intervals
TYPE=*keyword*	specifies the type of intervals (one-sided lower, one-sided upper, or two-sided)

Dictionary of Options

The following entries provide detailed descriptions of *options* in the INTERVALS statement.

ALPHA=*value-list*
specifies values of α, the probability or confidence associated with the interval. For example, the following statements tabulate the default intervals at probability or confidence levels of $\alpha = 0.05$, $\alpha = 0.10$, $\alpha = 0.15$, and $\alpha = 0.20$:

```
proc capability data=steel;
   intervals width / alpha = 0.05 0.10 0.15 0.20;
run;
```

Note that some references use $\gamma = 1 - \alpha$ to denote probability or confidence levels. Values for the ALPHA= option must be between 0.00001 to 0.99999. By default, values of 0.01, 0.05, and 0.10 are used.

K=*value-list*
lists values of k for prediction intervals. Default *values* of 1, 2, and 3 are used for the prediction interval for k future observations and for the prediction interval for the mean of k future observations. Default *values* of 2 and 3 are used for the prediction interval for the standard deviation of k future observations. The *values* must be integers.

METHODS=*indices*
METHOD=*indices*
specifies which intervals are computed. The *indices* can range from 1 to 6, and they correspond to the intervals described in Table 13.2.

Table 13.2. Intervals Computed for METHOD=*Index*

Index	Statistical Interval
1	approximate simultaneous prediction interval for k future observations
2	prediction interval for the mean of k future observations
3	approximate statistical tolerance interval that contains at least proportion p of the population
4	confidence interval for the population mean
5	prediction interval for the standard deviation of k future observations
6	confidence interval for the population standard deviation

For example, the following statements tabulate confidence limits for the population mean (METHOD=4) and confidence limits for the population standard deviation (METHOD=6):

```
proc capability data=steel;
   intervals width / methods=4 6;
run;
```

Formulas for the intervals are given in "Methods for Computing Statistical Intervals" on page 386. By default, the procedure computes all six intervals.

NOPRINT
suppresses the tables produced by default. This option is useful when you only want to save the interval information in an OUTINTERVALS= data set.

OUTINTERVALS=*SAS-data-set*
OUTINTERVAL=*SAS-data-set*
OUTINT=*SAS-data-set*
specifies an output SAS data set containing the intervals and related information. For example, the following statements create a data set named INTS containing intervals for the variable WIDTH:

```
proc capability data=steel;
   intervals width / outintervals=ints;
run;
```

See "OUTINTERVALS= Data Set" on page 389 for details.

P=*value-list*
lists values of p for the tolerance intervals. These values must be between 0.00001 to 0.99999. Note that the P= option applies only to the tolerance intervals (METHODS=3). By default, values of 0.90, 0.95, and 0.99 are used.

TYPE=LOWER | UPPER | TWOSIDED
determines whether the intervals computed are one-sided lower, one-sided upper, or two-sided intervals, respectively. See "Computing One-Sided Lower Prediction Limits" on page 382 for an example. The default interval type is TWOSIDED.

Details

This section provides details on the following topics:

- formulas for statistical intervals
- OUTINTERVALS= data sets

Methods for Computing Statistical Intervals

The formulas for statistical intervals given in this section use the following notation:

Notation	Definition
n	number of nonmissing values for a variable
$\bar{X}$	mean of variable
s	standard deviation of variable
z_α	$100\alpha^{\text{th}}$ percentile of the standard normal distribution
$t_\alpha(\nu)$	$100\alpha^{\text{th}}$ percentile of the central t distribution with ν degrees of freedom
$t'_\alpha(\delta,\nu)$	$100\alpha^{\text{th}}$ percentile of the noncentral t distribution with noncentrality parameter δ and ν degrees of freedom
$F_\alpha(\nu_1,\nu_2)$	$100\alpha^{\text{th}}$ percentile of the F distribution with ν_1 degrees of freedom in the numerator and ν_2 degrees of freedom in the denominator
$\chi^2_\alpha(\nu)$	$100\alpha^{\text{th}}$ percentile of the χ^2 distribution with ν degrees of freedom.

The values of the variable are assumed to be independent and normally distributed. The intervals are computed using the degrees of freedom as the divisor for the standard deviation s. This divisor corresponds to the default of VARDEF=DF in the PROC CAPABILITY statement. If you specify another value for the VARDEF= option, intervals are not computed.

You select the intervals to be computed with the METHODS= option. The next six sections give computational details for each of the METHODS= options.

METHODS=1

This requests an approximate simultaneous prediction interval for k future observations. Two-sided intervals are computed using the conservative approximations

$$\text{Lower Limit} = \bar{X} - t_{1-\frac{\alpha}{2k}}(n-1)s\sqrt{1+\tfrac{1}{n}}$$

$$\text{Upper Limit} = \bar{X} + t_{1-\frac{\alpha}{2k}}(n-1)s\sqrt{1+\tfrac{1}{n}}$$

One-sided limits are computed using the conservative approximation

$$\text{Lower Limit} = \bar{X} - t_{1-\frac{\alpha}{k}}(n-1)s\sqrt{1+\frac{1}{n}}$$

$$\text{Upper Limit} = \bar{X} + t_{1-\frac{\alpha}{k}}(n-1)s\sqrt{1+\frac{1}{n}}$$

Hahn (1970c) states that these approximations are satisfactory except for combinations of small n, large k, and large α. Refer also to Hahn (1969 and 1970a) and Hahn and Meeker (1991).

METHODS=2

This requests a prediction interval for the mean of k future observations. Two-sided intervals are computed as

$$\text{Lower Limit} = \bar{X} - t_{1-\frac{\alpha}{2}}(n-1)s\sqrt{\frac{1}{k}+\frac{1}{n}}$$

$$\text{Upper Limit} = \bar{X} + t_{1-\frac{\alpha}{2}}(n-1)s\sqrt{\frac{1}{k}+\frac{1}{n}}$$

One-sided limits are computed as

$$\text{Lower Limit} = \bar{X} - t_{1-\alpha}(n-1)s\sqrt{\frac{1}{k}+\frac{1}{n}}$$

$$\text{Upper Limit} = \bar{X} + t_{1-\alpha}(n-1)s\sqrt{\frac{1}{k}+\frac{1}{n}}$$

METHODS=3

This requests an approximate statistical tolerance interval that contains at least proportion p of the population. Two-sided intervals are approximated by

$$\text{Lower Limit} = \bar{X} - g(p\,;n\,;1-\alpha)s$$

$$\text{Upper Limit} = \bar{X} + g(p\,;n\,;1-\alpha)s$$

where $g(p\,;n\,;1-\alpha) = z_{\frac{1+p}{2}}(1+\frac{1}{2n})\sqrt{\frac{n-1}{\chi^2_\alpha(n-1)}}$.

Exact one-sided limits are computed as

$$\text{Lower Limit} = \bar{X} - g'(p\,;n\,;1-\alpha)s$$

$$\text{Upper Limit} = \bar{X} + g'(p\,;n\,;1-\alpha)s$$

where $g'(p\,;n\,;1-\alpha) = \frac{1}{\sqrt{n}}t'_{1-\alpha}(z_p\sqrt{n}, n-1)$.

In some cases (for example, if $z_p\sqrt{n}$ is large), $g'(p\,;n\,;1-\alpha)$ is approximated by

$$\frac{1}{a}\left(z_p + \sqrt{z_p^2 - ab}\right)$$

where $a = 1 - \frac{z_{1-\alpha}^2}{2(n-1)}$ and $b = z_p^2 - \frac{z_{1-\alpha}^2}{n}$.

Hahn (1970b) states that this approximation is "poor for very small n, especially for large p and large $1-\alpha$, and is not advised for $n < 8$." Refer also to Hahn and Meeker (1991).

METHODS=4

This requests a confidence interval for the population mean. Two-sided intervals are computed as

$$\text{Lower Limit} = \bar{X} - t_{1-\frac{\alpha}{2}}(n-1)\frac{s}{\sqrt{n}}$$

$$\text{Upper Limit} = \bar{X} + t_{1-\frac{\alpha}{2}}(n-1)\frac{s}{\sqrt{n}}$$

One-sided limits are computed as

$$\text{Lower Limit} = \bar{X} - t_{1-\alpha}(n-1)\frac{s}{\sqrt{n}}$$

$$\text{Upper Limit} = \bar{X} + t_{1-\alpha}(n-1)\frac{s}{\sqrt{n}}$$

METHODS=5

This requests a prediction interval for the standard deviation of k future observations. Two-sided intervals are computed as

$$\text{Lower Limit} = s\left(F_{1-\frac{\alpha}{2}}(n-1, k-1)\right)^{-\frac{1}{2}}$$

$$\text{Upper Limit} = s\left(F_{1-\frac{\alpha}{2}}(k-1, n-1)\right)^{\frac{1}{2}}$$

One-sided limits are computed as

$$\text{Lower Limit} = s\left(F_{1-\alpha}(n-1, k-1)\right)^{-\frac{1}{2}}$$

$$\text{Upper Limit} = s\left(F_{1-\alpha}(k-1, n-1)\right)^{\frac{1}{2}}$$

METHODS=6

This requests a confidence interval for the population standard deviation. Two-sided intervals are computed as

$$\text{Lower Limit} = s\sqrt{\frac{n-1}{\chi^2_{1-\frac{\alpha}{2}}(n-1)}}$$

$$\text{Upper Limit} = s\sqrt{\frac{n-1}{\chi^2_{\frac{\alpha}{2}}(n-1)}}$$

One-sided limits are computed as

$$\text{Lower Limit} = s\sqrt{\frac{n-1}{\chi^2_{1-\alpha}(n-1)}}$$

$$\text{Upper Limit} = s\sqrt{\frac{n-1}{\chi^2_{\alpha}(n-1)}}$$

OUTINTERVALS= Data Set

Each INTERVALS statement can create an output data set specified with the OUTINTERVALS= option. The OUTINTERVALS= data set contains statistical intervals and related parameters.

The number of observations in the OUTINTERVALS= data set depends on the number of variables analyzed, the number of tests specified, and the results of the tests. The OUTINTERVALS= data set is constructed as follows:

- The OUTINTERVALS= data set contains a group of observations for each variable analyzed.
- Each group contains one or more observations for each interval you specify with the METHODS= option. The actual number depends upon the number of combinations of the ALPHA=, K=, and P= values.

The following variables are saved in the OUTINTERVALS= data set:

Variable	Description
ALPHA	value of α associated with the intervals
_K_	value of K= for the prediction intervals
LOWER	lower endpoint of interval
METHOD	interval index (1–6)
_P_	value of P= for the tolerance intervals
TYPE	type of interval (ONESIDED or TWOSIDED)
UPPER	upper endpoint of interval
VAR	variable name

If you use a BY statement, the BY variables are also saved in the OUTINTERVALS= data set.

ODS Tables

The following table summarizes the ODS tables that you can request with the INTERVALS statement.

Table 13.3. ODS Tables Produced with the INTERVALS Statement

Table Name	Description	Option
Intervals1	prediction interval for future observations	METHODS=1
Intervals2	prediction interval for mean	METHODS=2
Intervals3	tolerance interval for proportion of population	METHODS=3
Intervals4	confidence limits for mean	METHODS=4
Intervals5	prediction interval for standard deviation	METHODS=5
Intervals6	confidence limits for standard deviation	METHODS=6

Chapter 14
OUTPUT Statement

Chapter Contents

OVERVIEW 393

GETTING STARTED 393
Saving Summary Statistics in an Output Data Set 393
Saving Percentiles in an Output Data Set 395

SYNTAX 396
Summary of Keywords 398

DETAILS 401
OUT= Data Set 401

EXAMPLES 402
Example 14.1. Computing Nonstandard Capability Indices 402
Example 14.2. Approximate Confidence Limits for Cpk 404

Chapter 14
OUTPUT Statement

Overview

You can use the OUTPUT statement to save summary statistics in a SAS data set. This information can then be used to create customized reports or to save historical information about a process.

You can use options in the OUTPUT statement to

- specify the statistics to save in the output data set
- specify the name of the output data set
- compute and save percentiles not automatically computed by the CAPABILITY procedure

Getting Started

This section introduces the OUTPUT statement with simple examples that illustrate commonly used options. Complete syntax for the OUTPUT statement is presented in the "Syntax" section on page 396, and advanced examples are given in the "Examples" section on page 402.

Saving Summary Statistics in an Output Data Set

See CAPOUT1 in the SAS/QC Sample Library

An automobile manufacturer producing seat belts saves summary information in an output data set using the CAPABILITY procedure. The following statements create the data set BELTS, which contains the breaking strengths (STRENGTH) and widths (WIDTH) of a sample of 50 belts:

```
data belts;
   label strength = 'Breaking Strength (lb/in)'
         width    = 'Width in Inches';
   input strength width @@;
datalines;
1243.51  3.036  1221.95  2.995  1131.67  2.983  1129.70  3.019
1198.08  3.106  1273.31  2.947  1250.24  3.018  1225.47  2.980
1126.78  2.965  1174.62  3.033  1250.79  2.941  1216.75  3.037
1285.30  2.893  1214.14  3.035  1270.24  2.957  1249.55  2.958
1166.02  3.067  1278.85  3.037  1280.74  2.984  1201.96  3.002
1101.73  2.961  1165.79  3.075  1186.19  3.058  1124.46  2.929
1213.62  2.984  1213.93  3.029  1289.59  2.956  1208.27  3.029
1247.48  3.027  1284.34  3.073  1209.09  3.004  1146.78  3.061
1224.03  2.915  1200.43  2.974  1183.42  3.033  1195.66  2.995
1258.31  2.958  1136.05  3.022  1177.44  3.090  1246.13  3.022
1183.67  3.045  1206.50  3.024  1195.69  3.005  1223.49  2.971
```

```
1147.47  2.944  1171.76  3.005  1207.28  3.065  1131.33  2.984
1215.92  3.003  1202.17  3.058
;
run;
```

The following statements produce two output data sets containing summary statistics:

```
proc capability data=belts;
   var strength width;
   output out=means    mean=smean wmean;
   output out=strstats mean=smean std=sstd min=smin max=smax;
run;
```

Note that if you specify an OUTPUT statement, you must also specify a VAR statement. You can use multiple OUTPUT statements with a single procedure statement. Each OUTPUT statement creates a new data set. The OUT= option specifies the name of the output data set. In this case, two data sets, MEANS and STRSTATS, are created. See Figure 14.1 for a listing of MEANS and Figure 14.2 for a listing of STRSTATS.

Summary statistics are saved in an output data set by specifying *keyword=names* after the OUT= option. In the preceding statements, the first OUTPUT statement specifies the *keyword* MEAN followed by the *names* SMEAN and WMEAN. The second OUTPUT statement specifies the *keywords* MEAN, STD, MAX, and MIN, for which the *names* SMEAN, SSTD, SMAX, and SMIN are given.

The *keyword* specifies the statistic to be saved in the output data set, and the *names* determine the names for the new variables. The first *name* listed after a keyword contains that statistic for the first variable listed in the VAR statement; the second *name* contains that statistic for the second variable in the VAR statement, and so on.

Thus, the data set MEANS contains the mean of STRENGTH in a variable named SMEAN and the mean of WIDTH in a variable named WMEAN. The data set STRSTATS contains the mean, standard deviation, maximum value, and minimum value of STRENGTH in the variables SMEAN, SSTD, SMAX, and SMIN, respectively.

Obs	smean	wmean
1	1205.75	3.00584

Figure 14.1. Listing of the Output Data Set MEANS

Obs	smean	sstd	smax	smin
1	1205.75	48.3290	1289.59	1101.73

Figure 14.2. Listing of the Output Data Set STRSTATS

Saving Percentiles in an Output Data Set

See CAPOUT1 in the SAS/QC Sample Library

The CAPABILITY procedure automatically computes the 1st, 5th, 10th, 25th, 75th, 90th, 95th, and 99th percentiles for each variable. You can save these percentiles in an output data set by specifying the appropriate keywords. For example, the following statements create an output data set named PCTLSTR containing the 5th and 95th percentiles of the variable STRENGTH:

```
proc capability data=belts noprint;
   var strength width;
   output out=pctlstr p5=p5str p95=p95str;
run;
```

The output data set PCTLSTR is listed in Figure 14.3.

Obs	p5str	p95str
1	1126.78	1284.34

Figure 14.3. Listing of the Output Data Set PCTLSTR

You can use the PCTLPTS=, PCTLPRE=, and PCTLNAME= options to save percentiles not automatically computed by the CAPABILITY procedure. For example, the following statements create an output data set named PCTLS containing the 20th and 40th percentiles of the variables STRENGTH and WIDTH:

```
proc capability data=belts noprint;
   var strength width;
   output out=pctls pctlpts  = 20 40
                    pctlpre  = S W
                    pctlname = pct20 pct40;
run;
```

The PCTLPTS= option specifies the percentiles to compute (in this case, the 20th and 40th percentiles). The PCTLPRE= and PCTLNAME= options build the names for the variables containing the percentiles. The PCTLPRE= option gives prefixes for the new variables, and the PCTLNAME= option gives a suffix to add to the prefix. Note that if you use the PCTLPTS= specification, you must also use the PCTLPRE= specification. For details on these options, see the "Syntax" section on page 396.

The preceding OUTPUT statement saves the 20th and 40th percentiles of STRENGTH and WIDTH in the variables SPCT20, WPCT20, SPCT40, and WPCT40. The output data set PCTLS is listed in Figure 14.4.

Obs	Spct20	Spct40	Wpct20	Wpct40
1	1165.91	1199.26	2.9595	2.995

Figure 14.4. Listing of the Output Data Set PCTLS

Syntax

The syntax for the OUTPUT statement is as follows:

OUTPUT <**OUT=***SAS-data-set*> *keyword=names . . . keyword=names*>
PCTLPTS=*percentiles* **PCTLPRE=** *prefixes* <**PCTLNAME=***suffixes*>**;**

You can use any number of OUTPUT statements in the CAPABILITY procedure. Each OUTPUT statement creates a new data set containing the statistics specified in that statement. When you use the OUTPUT statement, you must also use the VAR statement. In addition, the OUTPUT statement must contain at least one of the following:

- a specification of the form *keyword=names*
- the PCTLPTS= and PCTLPRE= specifications

The components of the OUTPUT statement are described as follows.

keyword=names
specifies the statistics to include in the output data set and gives names to the new variables that contain the statistics. Specify a *keyword* for each desired statistic, an equal sign, and the *names* of the variables to contain the statistic.

In the output data set, the first variable listed after a keyword in the OUTPUT statement contains the statistic for the first variable listed in the VAR statement; the second variable contains the statistic for the second variable in the VAR statement, and so on. The list of *names* following the equal sign can be shorter than the list of variables in the VAR statement. In this case, the procedure uses the *names* in the order in which the variables are listed in the VAR statement. Consider the following example:

```
proc capability noprint;
   var length width height;
   output out=summary mean=mlength mwidth;
run;
```

The variables MLENGTH and MWIDTH contain the means for LENGTH and WIDTH. The mean for HEIGHT is computed by the procedure but is not saved in the output data set. See "Summary of Keywords" on page 398 for tables of available keywords and the statistics they represent. Formulas for selected statistics are provided in the "Details" section beginning on page 187.

OUT=*SAS-data-set*
specifies the name of the output data set. To create a permanent SAS data set, specify a two-level name. See *SAS Language Reference: Dictionary* for more information on permanent SAS data sets. For example, the previous statements create an output data set named SUMMARY. If the OUT= option is omitted, then by default the new data set is named using the DATAn convention.

PCTLPTS=*percentiles*
specifies *percentiles* that are not automatically computed by the procedure. The

CAPABILITY procedure automatically computes the 1^{st}, 5^{th}, 10^{th}, 25^{th}, 50^{th}, 75^{th}, 90^{th}, 95^{th}, and 99^{th} percentiles for the data. These can be saved in an output data set using *keyword=names* specifications. The PCTLPTS= option generates additional percentiles and outputs them to a data set; these additional percentiles are not printed.

If you use the PCTLPTS= option, you must also use the PCTLPRE= option to provide a prefix for the new variable names. For example, to create variables that contain the 20^{th}, 40^{th}, 60^{th}, and 80^{th} percentiles of LENGTH, use the following statements:

```
proc capability noprint;
   var length;
   output pctlpts=20 40 60 80 pctlpre=plen;
run;
```

This creates the variables PLEN20, PLEN40, PLEN60, and PLEN80, whose values are the corresponding percentiles of LENGTH. In addition to specifying name prefixes with the PCTLPRE= option, you can also use the PCTLNAME= option to create name suffixes for the new variables created by the PCTLPTS= option.

PCTLPRE=*prefixes*

specifies prefixes used to create variable names for percentiles requested with the PCTLPTS= option. The PCTLPRE= and PCTLPTS= options must be used together.

The procedure generates new variable names using the *prefix* and the percentile values. If the specified percentile is an integer, the variable name is simply the *prefix* followed by the value. For noninteger percentiles, an underscore replaces the decimal point in the variable name, and decimal values are truncated to one decimal place. For example, the following statements create the variables PWID20, PWID33_3, PWID66_6, and PWID80 for the 20^{th}, 33.33^{rd}, 66.67^{th}, and 80^{th} percentiles of WIDTH, respectively:

```
proc capability noprint;
   var width;
   output pctlpts=20 33.33 66.67 80 pctlpre=pwid;
run;
```

If you request percentiles for more than one variable, you should list prefixes in the same order in which the variables appear in the VAR statement. If combining the *prefix* and percentile value results in a name longer than 8 characters, the prefix is truncated so that the variable name is 8 characters. For example, the following statements compute the 80^{th} and 87.5^{th} percentiles for LENGTH and WIDTH and save the new variables PLENGT80, PLEN87_5, PWIDTH80, and PWID87_5 in the output data set:

```
proc capability noprint;
   var length width;
   output pctlpts=80 87.5 pctlpre=plength pwidth;
run;
```

PCTLNAME=*suffixes*

provides name *suffixes* for the new variables created by the PCTLPTS= option. These *suffixes* are appended to the *prefixes* you specify with the PCTLPRE= option, replacing the percentile values that are used as suffixes by default. List the *suffixes* in the same order in which you specify the percentiles. If you specify n *suffixes* with the PCTLNAME= option and m percentile values with the PCTLPTS= option, where $m > n$, the *suffixes* are used to name the first n percentiles, and the default names are used for the remaining $m - n$ percentiles. For example, consider the following statements:

```
proc capability;
   var length width height;
   output pctlpts  = 20 40
          pctlpre  = pl pw ph
          pctlname = twenty;
run;
```

The value TWENTY in the PCTLNAME= option is used for only the first percentile in the PCTLPTS= list. This suffix is appended to the values in the PCTLPRE= option to generate the new variable names PLTWENTY, PWTWENTY, and PHTWENTY, which contain the 20^{th} percentiles for LENGTH, WIDTH, and HEIGHT, respectively. Since a second PCTLNAME= suffix is not specified, variable names for the 40^{th} percentiles for LENGTH, WIDTH, and HEIGHT are generated using the prefixes and percentile values. Thus, the output data set contains the variables PLTWENTY, PL40, PWTWENTY, PW40, PHTWENTY, and PH40.

If combining the prefix you specify with the PCTLPRE= option and the suffix you specify with the PCTLNAME= option results in a name longer than eight characters, the prefix is truncated from the right so that the variable name is exactly eight characters. For example, the following statements add the variables PLENGMED and PWIDTMED to the output data set:

```
proc capability;
   var length width;
   output pctlpts  = 50
          pctlpre  = plength pwidth
          pctlname = med;
run;
```

Summary of Keywords

The following tables list all keywords available in the OUTPUT statement grouped by type. Formulas for selected statistics are given in the "Details" section beginning on page 187.

Table 14.1. Descriptive Statistics

Keyword	Description
KURTOSIS	kurtosis
MAX	largest (maximum) value
MEAN	mean
MEDIAN	median (50^{th} percentile)
MIN	smallest (minimum) value
MODE	most frequent value (if not unique, the smallest mode is used)
N	number of observations on which calculations are based
NMISS	number of missing values
NOBS	number of observations
RANGE	range
SKEWNESS	skewness
STD	standard deviation
SUM	sum
SUMWGT	sum of weights
VAR	variance

Table 14.2. Specification Limits and Related Statistics

Keyword	Description
LSL	lower specification limit
PCTGTR	percent of nonmissing observations greater than the upper specification limit
PCTLSS	percent of nonmissing observations less than the lower specification limit
TARGET	target value
USL	upper specification limit

Table 14.3. Capability Indices and Related Statistics

Keyword	Description
CP	capability index C_p
CPLCL	lower confidence limit for C_p
CPUCL	upper confidence limit for C_p
CPK	capability index C_{pk} (also denoted *CPK*)
CPKLCL	lower confidence limit for C_{pk}
CPKUCL	upper confidence limit for C_{pk}
CPL	capability index *CPL*
CPLLCL	lower confidence limit for CPL
CPLUCL	upper confidence limit for CPL
CPM	capability index C_{pm}
CPMLCL	lower confidence limit for C_{pm}
CPMUCL	upper confidence limit for C_{pm}
CPU	capability index *CPU*
CPULCL	lower confidence limit for CPU
CPUUCL	upper confidence limit for CPU
K	capability index k (also denoted K)

Table 14.4. Quantile Statistics

Keyword	Description
MEDIAN	median (50^{th} percentile)
P1	1^{st} percentile
P5	5^{th} percentile
P10	10^{th} percentile
P90	90^{th} percentile
P95	95^{th} percentile
P99	99^{th} percentile
Q1	lower quartile (25^{th} percentile)
Q3	upper quartile (75^{th} percentile)
QRANGE	interquartile range (Q3−Q1)

Table 14.5. Normality and Signed Rank Test Statistics

Keyword	Description
NORMAL	test statistic for normality
PNORMAL	p-value for normality test
SIGNRANK	signed rank statistic

Details

OUT= Data Set

The CAPABILITY procedure creates an OUT= data set for each OUTPUT statement. The new data set contains an observation for each combination of levels of the variables in the BY statement, or a single observation if you do not specify a BY statement. Thus, the number of observations in the new data set corresponds to the number of groups for which statistics are calculated. The variables in the new data set are as follows:

- variables in the BY statement. The values of these variables match the values in the corresponding BY group in the DATA= data set.
- variables in the ID statement. The values of these variables match those for the first observation in each BY group, or for the first observation in the data set if you do not specify a BY statement.
- variables created by selecting statistics in the OUTPUT statement. The values of the statistics are computed using all the nonmissing data, or statistics are computed for each BY group if you use a BY statement.
- variables created by requesting new percentiles with the PCTLPTS= option. The names of these new variables depend on the values of the PCTLPRE= and PCTLNAME= options.

If the output data set contains a percentile variable or a quartile variable, the percentile definition assigned with the PCTLDEF= option in the PROC CAPABILITY statement is recorded on the output data set label.

The values of variables requested with the statistics keywords CP, CPK, CPL, CPM, CPU, K, PCTGTR, and PCTLSS are missing unless you identify specification limits in a SPEC statement or in a SPEC= data set.

As an alternative to OUT= data sets, you can create an OUTTABLE= data set. The structure of the OUTTABLE= data set may be more appropriate when you are computing summary statistics and capability indices for multiple process variables. See "OUTTABLE= Data Set" on page 190.

Examples

This section provides additional examples of the OUTPUT statement.

Example 14.1. Computing Nonstandard Capability Indices

See CPCPMK in the SAS/QC Sample Library

In recent years, a number of process capability indices that have been proposed in the research literature are gradually being introduced in applications. As shown in this example, you can compute such indices in the DATA step after using the OUTPUT statement in the CAPABILITY procedure to save various summary statistics.

Hardness measurements (in scaled units) for 50 titanium samples are saved as values of the variable HARDNESS in the following SAS data set:

```
data titanium;
   label hardness = 'Hardness Measurement';
   input hardness @@;
datalines;
1.38  1.49  1.43  1.60  1.59
1.34  1.44  1.64  1.83  1.57
1.45  1.74  1.61  1.39  1.63
1.73  1.61  1.35  1.51  1.47
1.46  1.41  1.56  1.40  1.58
1.43  1.53  1.53  1.58  1.62
1.58  1.46  1.26  1.57  1.41
1.53  1.36  1.63  1.36  1.66
1.49  1.55  1.67  1.41  1.39
1.75  1.37  1.36  1.86  1.49
;
```

The target value for hardness is 1.6, and the lower and upper specification limits are 0.8 and 2.4, respectively. The samples are produced by an in-control process, and the measurements are assumed to be normally distributed.

The following statements use the OUTPUT statement to save various descriptive statistics and an estimate of the index C_{pm} in a data set named INDICES:

```
proc capability data=titanium noprint;
   var hardness;
   specs lsl=0.8 target=1.6 usl=2.4;
   output out=indices
      n       = n
      mean    = avg
      std     = std
      var     = var
      lsl     = lsl
      target  = t
      usl     = usl
      pnormal = pnormal
      cpm     = cpm ;
run;
```

In addition to C_{pm}, you want to report an estimate for the index C_{pmk}, which is defined as follows:

$$C_{pmk} = \frac{d - |\mu - m|}{3\sqrt{\sigma^2 + (\mu - T)^2}}$$

where $d = (\text{USL} - \text{LSL})/2$, $m = (\text{USL} + \text{LSL})/2$, and μ and σ are the mean and standard deviation of the normal distribution. Refer to Section 3.6 of Kotz and Johnson (1993). A natural estimator for C_{pmk} is

$$\hat{C}_{pmk} = \frac{d - |\overline{X} - m|}{3\sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - T)^2}}$$

The following statements compute this estimate:

```
data indices;
   set indices;
   d    = 0.5*( USL - LSL );
   m    = 0.5*( USL + LSL );
   num  = d - abs( avg - m );
   den  = 3 * sqrt( (n-1)*var/n + (avg-t)*(avg-t) );
   cpmk = num/den;
run;


title "Capability Analysis of Titanium Hardness";
proc print data=indices noobs;
   var n avg std lsl t usl cpm cpmk pnormal;
run;
```

The results are listed in Output 14.1.1.

Output 14.1.1. Computation of C_{pmk}

Capability Analysis of Titanium Hardness

n	avg	std	lsl	t	usl	cpm	cpmk	pnormal
50	1.5212	0.13295	0.8	1.6	2.4	1.72545	1.56713	0.25111

Note that the p-value for the Kolmogorov-Smirnov test of normality is 0.27693, indicating that the assumption of normality is justified.

The following statements also compute an estimate of the index C_{pm} using the SPECIALINDICES option:

```
proc capability data=titanium specialindices;
   var hardness;
   specs lsl=0.8 target=1.6 usl=2.4;
run;
```

Output 14.1.2. Computation of C_{pmk} using the SPECIALINDICES option

The CAPABILITY Procedure
Variable: hardness (Hardness Measurement)

Process Capability Indices

Index	Value	95% Confidence Limits	
Cp	2.005745	1.609575	2.401129
CPL	1.808179	1.438675	2.175864
CPU	2.203311	1.757916	2.646912
Cpk	1.808179	1.438454	2.177904
Cpm	1.725446	1.410047	2.066027

Example 14.2. Approximate Confidence Limits for Cpk

See CPKCON3 in the SAS/QC Sample Library

This example illustrates how you can use the OUTPUT statement to compute confidence limits for the capability index C_{pk}.

You can request the approximate confidence limits given by Bissell (1990) with the keywords CPKLCL and CPKUCL in the OUTPUT statement. However, this is not the only method that has been proposed for computing confidence limits for C_{pk}. Zhang, Stenback, and Wardrop (1990), referred to here as ZSW, proposed approximate confidence limits of the form

$$\widehat{C}_{pk} \pm k\widehat{\sigma}_{pk}$$

where $\widehat{\sigma}_{pk}$ is an estimator of the standard deviation of $\widehat{C}_{pk}$. Equation (8) of ZSW provides an approximation to the variance of $\widehat{C}_{pk}$ from which one can obtain $100\gamma\%$ confidence limits for C_{pk} as

$$\begin{aligned}
\text{LCL} &= \widehat{C}_{pk}\left[1 - \Phi^{-1}((1-\gamma)/2)\sqrt{\frac{n-1}{n-3} - \frac{(n-1)\Gamma^2((n-2)/2)}{2\Gamma^2((n-1)/2)}}\,\right] \\
\text{UCL} &= \widehat{C}_{pk}\left[1 + \Phi^{-1}(1-(1-\gamma)/2)\sqrt{\frac{n-1}{n-3} - \frac{(n-1)\Gamma^2((n-2)/2)}{2\Gamma^2((n-1)/2)}}\,\right]
\end{aligned}$$

This assumes that $\widehat{C}_{pk}$ is normally distributed. You can also compute approximate confidence limits based on equation (6) of ZSW, which provides an exact expression for the variance of $\widehat{C}_{pk}$.

The following program uses the methods of Bissell (1990) and ZSW to compute approximate confidence limits for C_{pk} for the variable HARDNESS in the data set TITANIUM (see page 402).

```
proc capability data=titanium noprint;
   var hardness;
   specs lsl=0.8 usl=2.4 gamma=0.95;
   output out=summary
       n    = n
       mean = mean
       std  = std
       lsl  = lsl
       usl  = usl
       cpk  = cpk
       cpklcl = cpklcl
       cpkucl = cpkucl
       cpl  = cpl
       cpu  = cpu ;

data summary;
   set summary;
   length method $ 16;

   method = "Bissell";
   lcl = cpklcl;
   ucl = cpkucl;
   output;

   * Assign confidence level;
   level = 0.95;
   aux   = probit( 1 - (1-level)/2 );

   method = "ZSW Equation 6";
   zsw = log(0.5*n-0.5)
         + ( 2*(lgamma(0.5*n-1)-lgamma(0.5*n-0.5)) );
   zsw = sqrt((n-1)/(n-3)-exp(zsw));
   lcl = cpk*(1-aux*zsw);
   ucl = cpk*(1+aux*zsw);
   output;

   method = "ZSW Equation 8";
   ds = 3*(cpu+cpl)/2;
   ms = 3*(cpl-cpu)/2;
   f1 = (1/3)*sqrt((n-1)/2)*gamma((n-2)/2)*(1/gamma((n-1)/2));
   f2 = sqrt(2/n)*(1/gamma(0.5))*exp(-n*0.5*ms*ms);
   f3 = ms*(1-(2*probnorm(-sqrt(n)*ms)));
   ex = f1*(ds-f2-f3);
   sd = ((n-1)/(9*(n-3)))*(ds**2-(2*ds*(f2+f3))+ms**2+(1/n));
   sd = sd-(ex*ex);
   sd = sqrt(sd);
   lcl = cpk-aux*sd;
   ucl = cpk+aux*sd;
   output;
run;
```

```
title "Approximate 95% Confidence Limits for Cpk";
proc print data = summary noobs;
   var method lcl cpk ucl;
run;
```

The results are shown in Output 14.2.1.

Output 14.2.1. Approximate Confidence Limits for C_{pk}

Approximate 95% Confidence Limits for Cpk

method	lcl	cpk	ucl
Bissell	1.43845	1.80818	2.17790
ZSW Equation 6	1.43596	1.80818	2.18040
ZSW Equation 8	1.42419	1.80818	2.19217

Note that there is fairly close agreement in the three methods.

You can display the confidence limits computed using Bissell's approach on plots produced by the CAPABILITY procedure by specifying the keywords CPKLCL and CPKUCL in the INSET statement.

The following statements also compute an estimate of the index C_{pk} along with approximate limits using the SPECIALINDICES option:

```
proc capability data=titanium specialindices;
   var hardness;
   specs lsl=0.8 usl=2.4 gamma=0.95;
run;
```

Output 14.2.2. Approximate Confidence Limits for C_{pk} using the SPECIALINDICES option

The CAPABILITY Procedure
Variable: hardness (Hardness Measurement)

Process Capability Indices

Index	Value	95% Confidence Limits	
Cp	2.005745	1.609575	2.401129
CPL	1.808179	1.438675	2.175864
CPU	2.203311	1.757916	2.646912
Cpk	1.808179	1.438454	2.177904

Chapter 15
PPPLOT Statement

Chapter Contents

OVERVIEW 409

GETTING STARTED 410
Creating a Normal Probability-Probability Plot 410

SYNTAX 411
Summary of Options 412
Dictionary of Options 415

DETAILS 423
Construction and Interpretation of P-P Plots 423
Comparison of P-P Plots and Q-Q Plots 425
Summary of Theoretical Distributions 426
Specification of Symbol Markers 427
Specification of the Distribution Reference Line 427

Chapter 15
PPPLOT Statement

Overview

The PPPLOT statement creates a probability-probability plot (also referred to as a P-P plot or percent plot), which compares the empirical cumulative distribution function (ecdf) of a variable with a specified theoretical cumulative distribution function such as the normal. If the two distributions match, the points on the plot form a linear pattern that passes through the origin and has unit slope. Thus, you can use a P-P plot to determine how well a theoretical distribution models a set of measurements.

You can specify one of the following theoretical distributions with the PPPLOT statement:

- beta
- exponential
- gamma
- lognormal
- normal
- Weibull

You can use options in the PPPLOT statement to

- specify or estimate parameters for the theoretical distribution
- request graphical enhancements

Note: Probability-probability plots should not be confused with probability plots, which compare a set of ordered measurements with *percentiles* from a specified distribution. You can create probability plots with the PROBPLOT statement.

Getting Started

The following example illustrates the basic syntax of the PPPLOT statement. For complete details of the PPPLOT statement, see the "Syntax" section on page 411.

Creating a Normal Probability-Probability Plot

See CAPPP1 in the SAS/QC Sample Library

The distances between two holes cut into 50 steel sheets are measured and saved as values of the variable DISTANCE in the following data set:*

```
data sheets;
   input distance @@;
   label distance='Hole Distance in cm';
   datalines;
 9.80 10.20 10.27  9.70  9.76
10.11 10.24 10.20 10.24  9.63
 9.99  9.78 10.10 10.21 10.00
 9.96  9.79 10.08  9.79 10.06
10.10  9.95  9.84 10.11  9.93
10.56 10.47  9.42 10.44 10.16
10.11 10.36  9.94  9.77  9.36
 9.89  9.62 10.05  9.72  9.82
 9.99 10.16 10.58 10.70  9.54
10.31 10.07 10.33  9.98 10.15
;
run;
```

The cutting process is in statistical control. As a preliminary step in a capability analysis of the process, it is decided to check whether the distances are normally distributed. The following statements create a P-P plot, shown in Figure 15.1, which is based on the normal distribution with mean $\mu = 10$ and standard deviation $\sigma = 0.3$:

```
symbol v=plus;
title 'Normal Probability-Probability Plot for Hole Distance';
proc capability data=sheets noprint;
   ppplot distance / normal(mu=10 sigma=0.3 color=black)
                     square;
run;
```

The NORMAL option in the PPPLOT statement requests a P-P plot based on the normal cumulative distribution function, and the MU= and SIGMA= *normal-options* specify μ and σ. Note that a P-P plot is always based on a *completely specified* distribution, in other words, a distribution with specific parameters. In this example, if you did not specify the MU= and SIGMA= *normal-options*, the sample mean and sample standard deviation would be used for μ and σ.

*These data are also used to create Q-Q plots in Chapter 17, "QQPLOT Statement." See pages 465–466, 479–480, and 500.

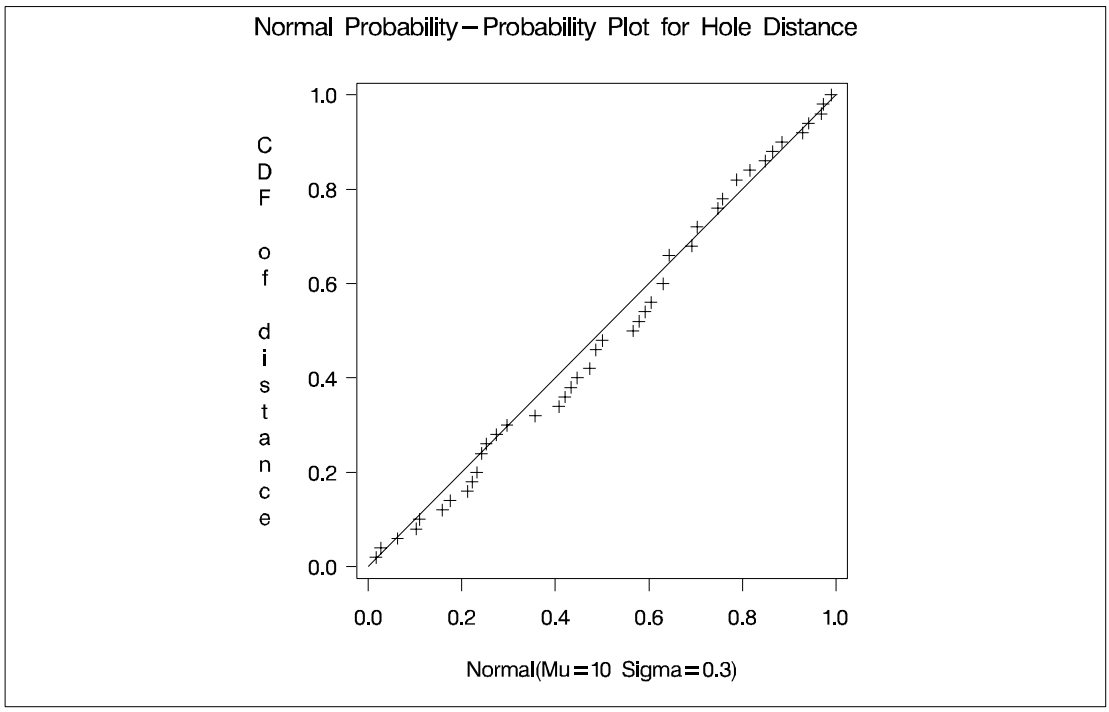


Figure 15.1. Normal P-P Plot with Diagonal Reference Line

The linearity of the pattern in Figure 15.1 is evidence that the measurements are normally distributed with mean 10 and standard deviation 0.3. The COLOR= *normal-option* specifies the color for the diagonal reference line, and the SQUARE option displays the plot in a square format.

Syntax

The syntax for the PPPLOT statement is as follows:

PPPLOT <*variables* > < **/** *options* >**;**

You can specify the keyword PP as an alias for PPPLOT, and you can use any number of PPPLOT statements in the CAPABILITY procedure. The components of the PPPLOT statement are described as follows.

variables
are the process variables for which to create P-P plots. If you specify a VAR statement, the *variables* must also be listed in the VAR statement. Otherwise, the *variables* can be any numeric variables in the input data set. If you do not specify a list of *variables*, then by default, the procedure creates a P-P plot for each variable listed in the VAR statement or for each numeric variable in the input data set if you do not specify a VAR statement. For example, each of the following PPPLOT statements produces two P-P plots, one for LENGTH and one for WIDTH:

```
proc capability data=measures;
   var length width;
   ppplot;
run;

proc capability data=measures;
   ppplot length width;
run;
```

options

specify the theoretical distribution for the plot or add features to the plot. If you specify more than one variable, the options apply equally to each variable. Specify all *options* after the slash (/) in the PPPLOT statement. You can specify only one option naming a distribution, but you can specify any number of other options. The distributions available are the beta, exponential, gamma, lognormal, normal, and Weibull. By default, the procedure produces a P-P plot based on the normal distribution.

In the following example, the NORMAL, MU= and SIGMA= options request a P-P plot based on the normal distribution with mean 10 and standard deviation 0.3. The SQUARE option displays the plot in a square frame, and the CTEXT= option specifies the text color.

```
proc capability data=measures;
   ppplot length width / normal(mu=10 sigma=0.3)
                         square
                         ctext=blue;
run;
```

Summary of Options

The following tables list the PPPLOT statement options by function. For complete descriptions, see the "Dictionary of Options" section on page 415.

Distribution Options

Table 15.1 summarizes the options for requesting a specific theoretical distribution.

Table 15.1. Options for Specifying the Theoretical Distribution

BETA(*beta-options*)	specifies beta P-P plot
EXPONENTIAL(*exponential-options*)	specifies exponential P-P plot
GAMMA(*gamma-options*)	specifies gamma P-P plot
LOGNORMAL(*lognormal-options*)	specifies lognormal P-P plot
NORMAL(*normal-options*)	specifies normal P-P plot
WEIBULL(*Weibull-options*)	specifies Weibull P-P plot

Table 15.2 through Table 15.8 summarize options that specify distribution parameters and control the display of the diagonal distribution reference line. Specify these options in parentheses after the distribution option. For example, the following statements use the NORMAL option to request a normal P-P plot:

```
proc capability data=measures;
   ppplot length / normal(mu=10 sigma=0.3 color=red);
run;
```

The MU= and SIGMA= *normal-options* specify μ and σ for the normal distribution, and the COLOR= *normal-option* specifies the color for the line.

Table 15.2. Distribution Reference Line Options

COLOR=*color*	specifies color of distribution reference line
L=*linetype*	specifies line type of distribution reference line
NOLINE	suppresses the distribution reference line
SYMBOL='*character*'	specifies plotting character for line printer
W=n	specifies width of distribution reference line

Table 15.3. Beta-Options

ALPHA=*value*	specifies shape parameter α
BETA=*value*	specifies shape parameter β
SIGMA=*value*	specifies scale parameter σ
THETA=*value*	specifies lower threshold parameter θ

Table 15.4. Exponential-Options

SIGMA=*value*	specifies scale parameter σ
THETA=*value*	specifies threshold parameter θ

Table 15.5. Gamma-Options

ALPHA=*value*	specifies shape parameter α
SIGMA=*value*	specifies scale parameter σ
THETA=*value*	specifies threshold parameter θ

Table 15.6. Lognormal-Options

SIGMA=*value*	specifies shape parameter σ
THETA=*value*	specifies threshold parameter θ
ZETA=*value*	specifies scale parameter ζ

Table 15.7. Normal-Options

MU=*value*	specifies mean μ
SIGMA=*value*	specifies standard deviation σ

Table 15.8. Weibull-Options

C=*value*	specifies shape parameter c
SIGMA=*value*	specifies scale parameter σ
THETA=*value*	specifies threshold parameter θ

General Options

Table 15.9 through Table 15.11 list options that control the appearance of the plots.

Table 15.9. General Plot Layout Options

HREF=*value-list*	specifies reference lines perpendicular to the horizontal axis
HREFLABELS='*label1*' . . . '*labeln*'	specifies line labels for HREF= lines
NOFRAME	suppresses frame around plotting area
SQUARE	displays P-P plot in square format
VREF=*value-list*	specifies reference lines perpendicular to the vertical axis
VREFLABELS='*label1*' . . . '*labeln*'	specifies line labels for VREF= lines

Table 15.10. Options to Enhance Plots Produced On Line Printers

HREFCHAR='*character*'	specifies line character for HREF= lines
NOOBSLEGEND	suppresses legend for hidden points
PPSYMBOL='*character*'	specifies character for plotted points
VREFCHAR='*character*'	specifies line character for VREF= lines

Table 15.11. Options to Enhance Plots Produced On Graphics Devices

ANNOTATE=*SAS-data-set*	provides an annotate data set
CAXIS=*color*	specifies color for axis
CFRAME=*color*	specifies color for frame
CHREF=*color*	specifies color for HREF= lines
CTEXT=*color*	specifies color for text
CVREF=*color*	specifies color for VREF= lines
DESCRIPTION='*string*'	specifies description for plot in graphics catalog
FONT=*font*	specifies software font for text
HAXIS=*name*	identifies AXIS statement for horizontal axis
HMINOR=*n*	specifies number of minor tick marks on horizontal axis
LHREF=*linetype*	specifies line type for HREF= lines
LVREF=*linetype*	specifies line type for VREF= lines
NAME='*string*'	specifies name for plot in graphics catalog
VAXIS=*name*	identifies AXIS statement for vertical axis
VMINOR=*value*	specifies number of minor tick marks on vertical axis

Dictionary of Options

The following entries provide detailed descriptions of options for the PPPLOT statement. The marginal notes *Graphics* and *Line Printer* identify options that apply to graphics devices and line printers, respectively.

ALPHA=*value*
specifies the shape parameter α ($\alpha > 0$) for P-P plots requested with the BETA and GAMMA options. For examples, see the entries for the BETA and GAMMA options.

ANNOTATE=*SAS-data-set*
ANNO=*SAS-data-set*
specifies an input data set containing annotate variables as described in *SAS/GRAPH Software: Reference*. You can use this data set to add features to the plot. The ANNOTATE= data set specified in the PPPLOT statement is used for all plots created by the statement. You can also specify an ANNOTATE= data set in the PROC CAPABILITY statement to enhance all plots created by the procedure; for more information, see "ANNOTATE= Data Sets" on page 189. *Graphics*

BETA<**(***beta-options***)**>
creates a beta P-P plot. To create the plot, the n nonmissing observations are ordered from smallest to largest:

$$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$$

The y-coordinate of the i^{th} point is the empirical cdf value $\frac{i}{n}$. The x-coordinate is the theoretical beta cdf value

$$B_{\alpha\beta}\left(\frac{x_{(i)}-\theta}{\sigma}\right) = \int_{\theta}^{x_{(i)}} \frac{(t-\theta)^{\alpha-1}(\theta+\sigma-t)^{\beta-1}}{B(\alpha,\beta)\sigma^{(\alpha+\beta-1)}}dt$$

where $B_{\alpha\beta}(\cdot)$ is the normalized incomplete beta function, $B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$, and

θ = lower threshold parameter
σ = scale parameter ($\sigma > 0$)
α = first shape parameter ($\alpha > 0$)
β = second shape parameter ($\beta > 0$)

You can specify α, β, σ, and θ with the ALPHA=, BETA=, SIGMA=, and THETA= *beta-options*, as illustrated in the following example:

```
proc capability data=measures;
   ppplot width / beta(theta=1 sigma=2 alpha=3 beta=4);
run;
```

If you do not specify values for these parameters, then by default, $\theta = 0$, $\sigma = 1$, and maximum likelihood estimates are calculated for α and β.

IMPORTANT: If the default unit interval (0,1) does not adequately describe the range of your data, then you should specify THETA=θ and SIGMA=σ so that your data fall in the interval $(\theta, \theta + \sigma)$.

If the data are beta distributed with parameters α, β, σ, and θ, then the points on the plot for ALPHA=α, BETA=β, SIGMA=σ, and THETA=θ tend to fall on or near the diagonal line $y = x$, which is displayed by default. Agreement between the diagonal line and the point pattern is evidence that the specified beta distribution is a good fit. You can specify the SCALE= option as an alias for the SIGMA= option and the THRESHOLD= option as an alias for the THETA= option.

BETA=*value*

specifies the shape parameter β ($\beta > 0$) for P-P plots requested with the BETA distribution option. See the preceding entry for the BETA distribution option for an example.

C=*value*

specifies the shape parameter c ($c > 0$) for P-P plots requested with the WEIBULL option. See the entry for the WEIBULL option for examples.

CAXIS=*color*
CAXES=*color*

Graphics

specifies the color for the axes. This option overrides any COLOR= specifications in an AXIS statement. The default is the first color in the device color list.

CFRAME=*color*
CFR=*color*

Graphics

specifies a fill color for the area enclosed by the axes and frame. By default, this area is not filled.

CHREF=*color*
CH=*color*

Graphics

specifies the color for reference lines requested by the HREF= option. The default is the first color in the device color list.

COLOR=*color*

Graphics

specifies the color for the diagonal reference line. For example, the following statements request a blue line:

```
proc capability data=measures;
   ppplot length / normal(mu=10 sigma=0.25 color=blue);
run;
```

The default is the first color in the device color list.

CTEXT=*color*

Graphics

specifies the color for tick mark values and axis labels. The default is the color specified for the CTEXT= option in the most recent GOPTIONS statement.

CVREF=*color*
CV=*color*

Graphics

specifies the color for reference lines requested by the VREF= option. The default is the first color in the device color list.

DESCRIPTION='*string*'
DES='*string*'
specifies a description, up to 40 characters, that appears in the PROC GREPLAY master menu. The default string is the variable name. *Graphics*

EXPONENTIAL<**(***exponential-options***)**>
EXP<**(***exponential-options***)**>
creates an exponential P-P plot. To create the plot, the n nonmissing observations are ordered from smallest to largest:

$$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$$

The y-coordinate of the i^{th} point is the empirical cdf value $\frac{i}{n}$. The x-coordinate is the theoretical exponential cdf value

$$F(x_{(i)}) = 1 - \exp\left(-\frac{x_{(i)}-\theta}{\sigma}\right)$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)

You can specify σ and θ with the SIGMA= and THETA= *exponential-options*, as illustrated in the following example:

```
proc capability data=measures;
   ppplot width / exponential(theta=1 sigma=2);
run;
```

If you do not specify values for these parameters, then by default, $\theta = 0$ and a maximum likelihood estimate is calculated for σ.

IMPORTANT: Your data must be greater than or equal to the lower threshold θ. If the default $\theta = 0$ is not an adequate lower bound for your data, specify θ with the THETA= option.

If the data are exponentially distributed with parameters σ and θ, the points on the plot for SIGMA=σ and THETA=θ tend to fall on or near the diagonal line $y = x$, which is displayed by default. Agreement between the diagonal line and the point pattern is evidence that the specified exponential distribution is a good fit. You can specify the SCALE= option as an alias for the SIGMA= option and the THRESHOLD= option as an alias for the THETA= option.

FONT=*font*
specifies a software font for horizontal and vertical reference line labels and axis labels. You can also specify fonts for axis labels in an AXIS statement. The FONT= font takes precedence over the FTEXT= font you specify in the GOPTIONS statement. Hardware characters are used by default. *Graphics*

GAMMA<**(***gamma-options***)**>
creates a gamma P-P plot. To create the plot, the n nonmissing observations are ordered from smallest to largest:

$$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$$

The y-coordinate of the i^{th} point is the empirical cdf value $\frac{i}{n}$. The x-coordinate is the theoretical gamma cdf value

$$G_\alpha\left(\frac{x_{(i)}-\theta}{\sigma}\right) = \int_\theta^{x_{(i)}} \frac{1}{\sigma\Gamma(\alpha)}\left(\frac{t-\theta}{\sigma}\right)^{\alpha-1}\exp\left(-\frac{t-\theta}{\sigma}\right)dt$$

where $G_\alpha(\cdot)$ is the normalized incomplete gamma function, and

$\theta =$ threshold parameter
$\sigma =$ scale parameter $(\sigma > 0)$
$\alpha =$ shape parameter $(\alpha > 0)$

You can specify α, σ, and θ with the ALPHA=, SIGMA=, and THETA= *gamma-options*, as illustrated in the following example:

```
proc capability data=measures;
   ppplot width / gamma(alpha=1 sigma=2 theta=3);
run;
```

If you do not specify values for these parameters, then by default, $\theta = 0$ and maximum likelihood estimates are calculated for α and σ.

IMPORTANT: Your data must be greater than or equal to the lower threshold θ. If the default $\theta = 0$ is not an adequate lower bound for your data, specify θ with the THETA= option.

If the data are gamma distributed with parameters α, σ, and θ, the points on the plot for ALPHA=α, SIGMA=σ, and THETA=θ tend to fall on or near the diagonal line $y = x$, which is displayed by default. Agreement between the diagonal line and the point pattern is evidence that the specified gamma distribution is a good fit. You can specify the SHAPE= option as an alias for the ALPHA= option, the SCALE= option as an alias for the SIGMA= option, and the THRESHOLD= option as an alias for the THETA= option.

HAXIS=*name*

Graphics

specifies the name of an AXIS statement describing the horizontal axis.

HMINOR=n

HM=n

Graphics

specifies the number of minor tick marks between each major tick mark on the horizontal axis. Minor tick marks are not labeled. The default is 0.

HREF=*value-list*

draws reference lines perpendicular to the horizontal axis at the values specified. See also the HREFCHAR=, CHREF=, and LHREF= options.

HREFCHAR='*character***'**

Line Printer

specifies the character used to form the reference lines requested by the HREF= option for a line printer. The default is the vertical bar (|).

HREFLABELS=*'label1' . . . 'labeln'*
HREFLABEL=*'label1' . . . 'labeln'*
HREFLAB=*'label1' . . . 'labeln'*
specifies labels for the reference lines requested by the HREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

L=*linetype*
specifies the line type for the diagonal distribution reference line. For example, *Graphics*

```
proc capability data=measures;
   ppplot length / normal(mu=10 sigma=0.25 l=2);
run;
```

The default is 1, which produces a solid line.

LHREF=*linetype*
LH=*linetype*
specifies the line type for reference lines requested by the HREF= option. The default is 2, which produces a dashed line. *Graphics*

LOGNORMAL<**(***lognormal-options***)**>
LNORM<**(***lognormal-options***)**>
creates a lognormal P-P plot. To create the plot, the n nonmissing observations are ordered from smallest to largest:

$$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$$

The y-coordinate of the i^{th} point is the empirical cdf value $\frac{i}{n}$. The x-coordinate is the theoretical lognormal cdf value

$$\Phi\left(\frac{\log(x_{(i)}-\theta)-\zeta}{\sigma}\right)$$

where $\Phi(\cdot)$ is the cumulative standard normal distribution function, and

$\theta =$ threshold parameter
$\zeta =$ scale parameter
$\sigma =$ shape parameter $(\sigma > 0)$

You can specify θ, ζ, and σ with the THETA=, ZETA=, and SIGMA= *lognormal-options*, as illustrated in the following example:

```
proc capability data=measures;
   ppplot width / lognormal(theta=1 zeta=2);
run;
```

If you do not specify values for these parameters, then by default, $\theta = 0$ and maximum likelihood estimates are calculated for σ and ζ.

IMPORTANT: Your data must be greater than the lower threshold θ. If the default $\theta = 0$ is not an adequate lower bound for your data, specify θ with the THETA= option.

If the data are lognormally distributed with parameters σ, θ, and ζ, the points on the plot for SIGMA=σ, THETA=θ, and ZETA=ζ tend to fall on or near the diagonal line $y = x$, which is displayed by default. Agreement between the diagonal line and the point pattern is evidence that the specified lognormal distribution is a good fit. You can specify the SHAPE= option as an alias for the SIGMA=option, the SCALE= option as an alias for the ZETA= option, and the THRESHOLD= option as an alias for the THETA= option.

LVREF=*linetype*
LV=*linetype*

Graphics

specifies the line type for reference lines requested by the VREF= option. The default is 2, which produces a dashed line.

MU=*value*

specifies the mean μ for a normal P-P plot requested with the NORMAL option. For examples, see Figure 15.1 on page 411, or Figure 15.2 on page 424 and Figure 15.3 on page 425. By default, the sample mean is used for μ.

NAME='*string* **'**

Graphics

specifies a name for the plot, up to eight characters, that appears in the PROC GREPLAY master menu. The default name is 'CAPABILI'.

NOFRAME

suppresses the frame around the subplot area.

NOLINE

suppresses the diagonal reference line.

NOOBSLEGEND
NOOBSL

Line Printer

suppresses the legend that indicates the number of hidden observations.

NORMAL<**(***normal-options* **)**>
NORM<**(***normal-options* **)**>

creates a normal P-P plot. By default, if you do not specify a distribution option, the procedure displays a normal P-P plot. To create the plot, the n nonmissing observations are ordered from smallest to largest:

$$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$$

The y-coordinate of the i^{th} point is the empirical cdf value $\frac{i}{n}$. The x-coordinate is the theoretical normal cdf value

$$\Phi\left(\frac{x_{(i)}-\mu}{\sigma}\right) = \int_{-\infty}^{x_{(i)}} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right) dt$$

where $\Phi(\cdot)$ is the cumulative standard normal distribution function, and

$\mu =$ location parameter or mean
$\sigma =$ scale parameter or standard deviation $(\sigma > 0)$

You can specify μ and σ with the MU= and SIGMA= *normal-options*, as illustrated in the following example:

```
proc capability data=measures;
   ppplot width / normal(mu=1 sigma=2);
run;
```

By default, the sample mean and sample standard deviation are used for μ and σ.

If the data are normally distributed with parameters μ and σ, the points on the plot for MU=μ and SIGMA=σ tend to fall on or near the diagonal line $y = x$, which is displayed by default. Agreement between the diagonal line and the point pattern is evidence that the specified normal distribution is a good fit. For an example, see Figure 15.1 on page 411.

PPSYMBOL='*character***'**
specifies the character used to plot the points when the P-P plot is produced on a line printer. The default is the plus sign (+). *Line Printer*

SCALE=*value*
is an alias for the SIGMA= option with the BETA, EXPONENTIAL, GAMMA, and WEIBULL options and an alias for the ZETA= option with the LOGNORMAL option. See the entries for the SIGMA= and ZETA= options.

SHAPE=*value*
is an alias for the ALPHA= option with the GAMMA option, for the SIGMA= option with the LOGNORMAL option, and for the C= option with the WEIBULL option. See the entries for the ALPHA=, C=, and SIGMA= options.

SIGMA=*value*
specifies the parameter σ, where $\sigma > 0$. When used with the BETA, EXPONENTIAL, GAMMA, NORMAL, and WEIBULL options, the SIGMA= option specifies the scale parameter. When used with the LOGNORMAL option, the SIGMA= option specifies the shape parameter. For an example of the SIGMA= option used with the NORMAL option, see Figure 15.1 on page 411.

SQUARE
displays the P-P plot in a square frame. The default is a rectangular frame. See Figure 15.1 on page 411 for an example.

SYMBOL='*character***'**
specifies the character used to plot the diagonal reference line for a line printer. The default character is the first letter of the distribution option keyword. *Line Printer*

THETA=*value*
specifies the lower threshold parameter θ for plots requested with the BETA, EXPONENTIAL, GAMMA, LOGNORMAL, and WEIBULL options.

THRESHOLD=*value*
is an alias for the THETA= option.

VAXIS=*name*
specifies the name of an AXIS statement describing the vertical axis. For an example, see Figure 15.2 on page 424 and Figure 15.3 on page 425. *Graphics*

VMINOR=n
VM=n

Graphics

specifies the number of minor tick marks between each major tick mark on the vertical axis. Minor tick marks are not labeled. The default is 0.

VREF=*value-list*

draws reference lines perpendicular to the vertical axis at the values specified. See the entries for the VREFCHAR=, CVREF=, and LVREF= options.

VREFCHAR='*character***'**

Line Printer

specifies the character used to form the reference lines requested by the VREF= option for a line printer. The default is the hyphen (-).

VREFLABELS='*label1' . . . 'labeln'*
VREFLABEL='*label1' . . . 'labeln'*
VREFLAB='*label1' . . . 'labeln'*

specifies labels for the reference lines requested by the VREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

W=n

Graphics

specifies the width in pixels for the diagonal reference line. Specify the W= option in parentheses following a distribution option keyword. For a similar syntax example, see the entry for the L= option. The default is 1.

WEIBULL<**(***Weibull-options***)**>
WEIB<**(***Weibull-options***)**>

creates a Weibull P-P plot. To create the plot, the n nonmissing observations are ordered from smallest to largest:

$$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$$

The y-coordinate of the i^{th} point is the empirical cdf value $\frac{i}{n}$. The x-coordinate is the theoretical Weibull cdf value

$$F(x_{(i)}) = 1 - \exp\left(-\left(\frac{x_{(i)}-\theta}{\sigma}\right)^c\right)$$

where

$\theta =$ threshold parameter
$\sigma =$ scale parameter $(\sigma > 0)$
$c =$ shape parameter $(c > 0)$

You can specify c, σ, and θ with the C=, SIGMA=, and THETA= *Weibull-options*, as illustrated in the following example:

```
proc capability data=measures;
   ppplot width / weibull(theta=1 sigma=2);
run;
```

If you do not specify values for these parameters, then by default $\theta = 0$ and maximum likelihood estimates are calculated for σ and c.

IMPORTANT: Your data must be greater than or equal to the lower threshold θ. If the default $\theta = 0$ is not an adequate lower bound for your data, you should specify θ with the THETA= option.

If the data are Weibull distributed with parameters c, σ, and θ, the points on the plot for C=c, SIGMA=σ, and THETA=θ tend to fall on or near the diagonal line $y = x$, which is displayed by default. Agreement between the diagonal line and the point pattern is evidence that the specified Weibull distribution is a good fit. You can specify the SHAPE= option as an alias for the C= option, the SCALE= option as an alias for the SIGMA= option, and the THRESHOLD= option as an alias for the THETA= option.

ZETA=*value*
specifies a value for the scale parameter ζ for lognormal P-P plots requested with the LOGNORMAL option.

Details

This section provides details on the following topics:

- construction and interpretation of P-P plots
- comparison of P-P plots with Q-Q plots
- distributions supported by the PPPLOT statement
- graphical enhancements of P-P plots

Construction and Interpretation of P-P Plots

A P-P plot compares the empirical cumulative distribution function (ecdf) of a variable with a specified theoretical cumulative distribution function $F(\cdot)$. The ecdf, denoted by $F_n(x)$, is defined as the proportion of nonmissing observations less than or equal to x, so that $F_n(x_{(i)}) = \frac{i}{n}$.

To construct a P-P plot, the n nonmissing values are first sorted in increasing order:

$$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$$

Then the i^{th} ordered value $x_{(i)}$ is represented on the plot by the point whose x-coordinate is $F(x_{(i)})$ and whose y-coordinate is $\frac{i}{n}$.

Like Q-Q plots and probability plots, P-P plots can be used to determine how well a theoretical distribution models a data distribution. If the theoretical cdf reasonably models the ecdf in all respects, including location and scale, the point pattern on the P-P plot is linear through the origin and has unit slope.

Unlike Q-Q and probability plots, P-P plots are not invariant to changes in location and scale. For example, the data in the "Getting Started" section on page 410 are reasonably described by a normal distribution with mean 10 and standard deviation 0.3. It is instructive to display these data on normal P-P plots with a different mean and standard deviation, as created by the following statements:

See CAPPP2 in the SAS/QC Sample Library

```
symbol v=plus;
title 'Normal Probability-Probability Plot for Hole Distance';
proc capability data=sheets noprint;
   ppplot distance / normal(mu=9.5 sigma=0.3 color=black)
                     square
                     vaxis=axis1;


   ppplot distance / normal(mu=10 sigma=0.5 color=black)
                     square
                     vaxis = axis1;
   axis1 label=(a=90 r=0);
run;
```

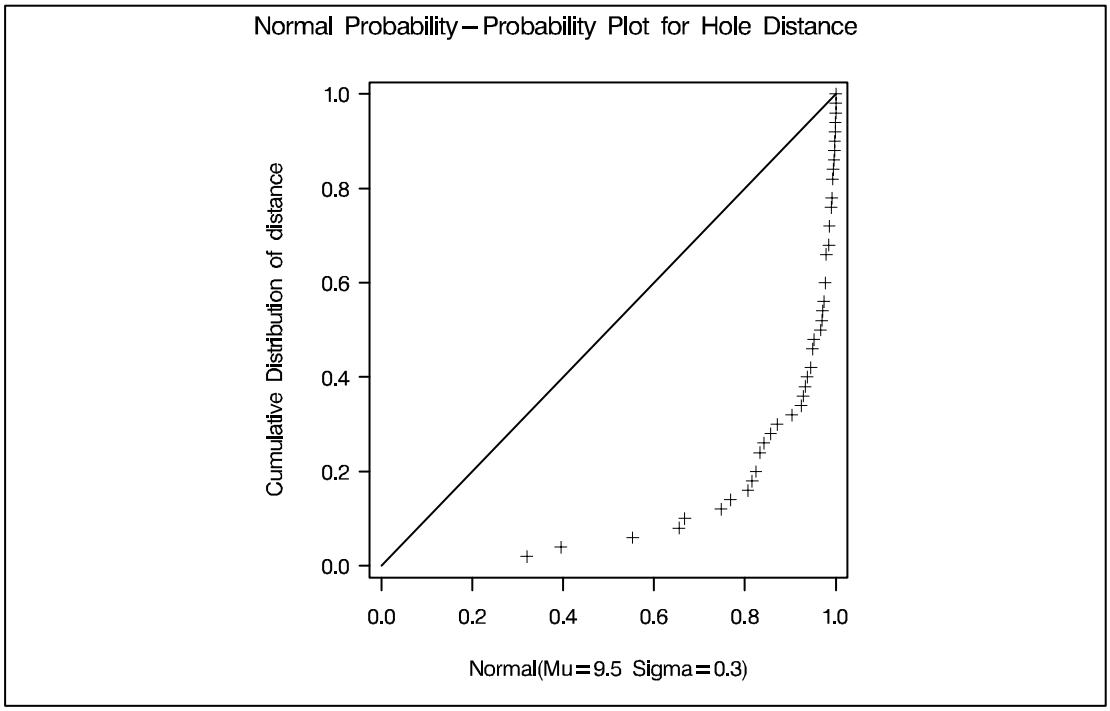


Figure 15.2. Normal P-P Plot with Mean Specified Incorrectly

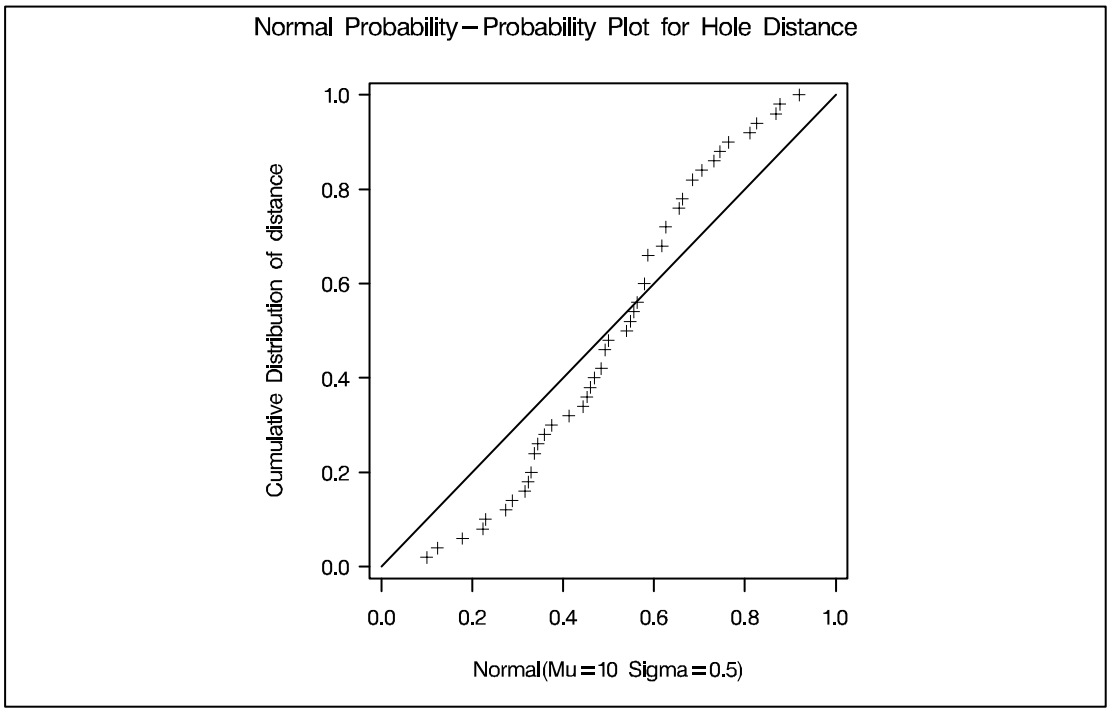


Figure 15.3. Normal P-P Plot with Standard Deviation Specified Incorrectly

Specifying a mean of 9.5 instead of 10 results in the plot shown in Figure 15.2, while specifying a standard deviation of 0.5 instead of 0.3 results in the plot shown in Figure 15.3. Both plots clearly reveal the model misspecification.

Comparison of P-P Plots and Q-Q Plots

A P-P plot compares the empirical cumulative distribution function of a data set with a specified theoretical cumulative distribution function $F(\cdot)$. A Q-Q plot compares the quantiles of a data distribution with the quantiles of a standardized theoretical distribution from a specified family of distributions. There are three important differences in the way P-P plots and Q-Q plots are constructed and interpreted:

- The construction of a Q-Q plot does not require that the location or scale parameters of $F(\cdot)$ be specified. The theoretical quantiles are computed from a standard distribution within the specified family. A linear point pattern indicates that the specified family reasonably describes the data distribution, and the location and scale parameters can be estimated visually as the intercept and slope of the linear pattern. In contrast, the construction of a P-P plot requires the location and scale parameters of $F(\cdot)$ to evaluate the cdf at the ordered data values.
- The linearity of the point pattern on a Q-Q plot is unaffected by changes in location or scale. On a P-P plot, changes in location or scale do not necessarily preserve linearity.

- On a Q-Q plot, the reference line representing a particular theoretical distribution depends on the location and scale parameters of that distribution, having intercept and slope equal to the location and scale parameters. On a P-P plot, the reference line for any distribution is always the diagonal line $y = x$.

Consequently, you should use a Q-Q plot if your objective is to compare the data distribution with a family of distributions that vary only in location and scale, particularly if you want to estimate the location and scale parameters from the plot.

An advantage of P-P plots is that they are discriminating in regions of high probability density, since in these regions the empirical and theoretical cumulative distributions change more rapidly than in regions of low probability density. For example, if you compare a data distribution with a particular normal distribution, differences in the middle of the two distributions are more apparent on a P-P plot than on a Q-Q plot.

For further details on P-P plots, refer to Gnanadesikan (1997) and Wilk and Gnanadesikan (1968).

Summary of Theoretical Distributions

You can use the PPPLOT statement to request P-P plots based on the theoretical distributions summarized in the following table:

Table 15.12. Distributions and Parameters

Family	Distribution Function $F(x)$	Range	Parameters		
			Location	Scale	Shape
Beta	$\int_{\theta}^{x} \frac{(t-\theta)^{\alpha-1}(\theta+\sigma-t)^{\beta-1}}{B(\alpha,\beta)\sigma^{(\alpha+\beta-1)}} dt$	$\theta < x < \theta + \sigma$	θ	σ	α, β
Exponential	$1 - \exp\left(-\frac{x-\theta}{\sigma}\right)$	$x \geq \theta$	θ	σ	
Gamma	$\int_{\theta}^{x} \frac{1}{\sigma\Gamma(\alpha)} \left(\frac{t-\theta}{\sigma}\right)^{\alpha-1} \exp\left(-\frac{t-\theta}{\sigma}\right) dt$	$x > \theta$	θ	σ	α
Lognormal	$\int_{\theta}^{x} \frac{1}{\sigma\sqrt{2\pi}(t-\theta)} \exp\left(-\frac{(\log(t-\theta)-\zeta)^2}{2\sigma^2}\right) dt$	$x > \theta$	θ	ζ	σ
Normal	$\int_{-\infty}^{x} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right) dt$	all x	μ	σ	
Weibull	$1 - \exp\left(-\left(\frac{x-\theta}{\sigma}\right)^{c}\right)$	$x > \theta$	θ	σ	c

You can request these distributions with the BETA, EXPONENTIAL, GAMMA, LOGNORMAL, NORMAL, and WEIBULL options, respectively. If you do not specify a distribution option, a normal P-P plot is created.

To create a P-P plot, you must provide all of the parameters for the theoretical distribution. If you do not specify parameters, then default values or estimates are substituted, as summarized by the following table:

Table 15.13. Defaults for Parameters

Family	Default Values	Estimated Values
Beta	$\theta = 0, \sigma = 1$	maximum likelihood estimates for α and β
Exponential	$\theta = 0$	maximum likelihood estimate for σ
Gamma	$\theta = 0$	maximum likelihood estimates for σ and α
Lognormal	$\theta = 0$	maximum likelihood estimates for σ and ζ
Normal	None	sample estimates for μ and σ
Weibull	$\theta = 0$	maximum likelihood estimates for σ and c

Specification of Symbol Markers

If you produce the P-P plot on a graphics device, you can use options in the SYMBOL1 statement to specify the appearance of the symbol marker for the points. The V= option specifies the symbol, the C= option specifies the color, and the H= option specifies the height. Refer to *SAS/GRAPH Software: Reference* for details concerning these options. If you produce the plot on a line printer, you can use the PPSYMBOL= option in the PPPLOT statement to specify the character used to plot the points.

Specification of the Distribution Reference Line

If you produce the P-P plot on a graphics device, you can control the color, type, and width of the diagonal distribution reference line by specifying the COLOR=, L=, and W= options in parentheses after the distribution option in the PPPLOT statement. Alternatively, you can control these features with the C=, L=, and W= options in the SYMBOL4 statement. Refer to *SAS/GRAPH Software: Reference* for details concerning these options. If you produce the plot on a line printer, you can specify the character used for the line with the SYMBOL= option enclosed in parentheses after the distribution option in the PPPLOT statement.

Chapter 16
PROBPLOT Statement

Chapter Contents

OVERVIEW 431

GETTING STARTED 431
Creating a Normal Probability Plot 432
Creating Lognormal Probability Plots 434

SYNTAX 437
Summary of Options 438
Dictionary of Options 441

DETAILS 454
Summary of Theoretical Distributions 454
SYMBOL Statement Options 456

EXAMPLES 457
Example 16.1. Displaying a Normal Reference Line 457
Example 16.2. Displaying a Lognormal Reference Line 459

Chapter 16
PROBPLOT Statement

Overview

The PROBPLOT statement creates a probability plot, which compares ordered values of a variable with percentiles of a specified theoretical distribution such as the normal. If the data distribution matches the theoretical distribution, the points on the plot form a linear pattern. Thus, you can use a probability plot to determine how well a theoretical distribution models a set of measurements.

You can specify one of the following theoretical distributions with the PROBPLOT statement:

- beta
- exponential
- gamma
- three-parameter lognormal
- normal
- two-parameter Weibull
- three-parameter Weibull

You can use options in the PROBPLOT statement to

- specify or estimate shape parameters for the theoretical distribution
- display a reference line corresponding to specified or estimated location and scale parameters for the theoretical distribution
- request graphical enhancements

Note: Probability plots are similar to Q-Q plots, which you can create with the QQPLOT statement (see Chapter 17, "QQPLOT Statement,"). Probability plots are preferable for graphical estimation of percentiles, whereas Q-Q plots are preferable for graphical estimation of distribution parameters and capability indices.

Getting Started

The following examples illustrate the basic syntax of the PROBPLOT statement. For complete details of the PROBPLOT statement, see the "Syntax" section on page 437. Advanced examples are provided on the "Examples" section on page 457.

Creating a Normal Probability Plot

See CAPPROB1 in the SAS/QC Sample Library

The diameters of 50 steel rods are measured and saved as values of the variable DISTANCE in the following data set:*

```
data rods;
   input diameter @@;
   label diameter='Diameter in mm';
   datalines;
  5.501  5.251  5.404  5.366  5.445
  5.576  5.607  5.200  5.977  5.177
  5.332  5.399  5.661  5.512  5.252
  5.404  5.739  5.525  5.160  5.410
  5.823  5.376  5.202  5.470  5.410
  5.394  5.146  5.244  5.309  5.480
  5.388  5.399  5.360  5.368  5.394
  5.248  5.409  5.304  6.239  5.781
  5.247  5.907  5.208  5.143  5.304
  5.603  5.164  5.209  5.475  5.223
  ;
run;
```

The process producing the rods is in statistical control, and as a preliminary step in a capability analysis of the process, you decide to check whether the diameters are normally distributed. The following statements create the normal probability plot shown in Figure 16.1:

```
symbol v=plus;
title 'Normal Probability Plot for Diameters';
proc capability data=rods noprint;
   probplot diameter;
run;
```

If you specify the LINEPRINTER option in the PROC CAPABILITY statement, the plot is created using a line printer, as shown in Figure 16.2. Note that the PROBPLOT statement creates a normal probability plot for DIAMETER by default.

The nonlinearity of the point pattern indicates a departure from normality. Since the point pattern is curved with slope increasing from left to right, a theoretical distribution that is skewed to the right, such as a lognormal distribution, should provide a better fit than the normal distribution. This possibility is explored in the next example.

*This data set is analyzed using quantile-quantile plots in Example 17.1 on page 491 and Example 17.2 on page 492.

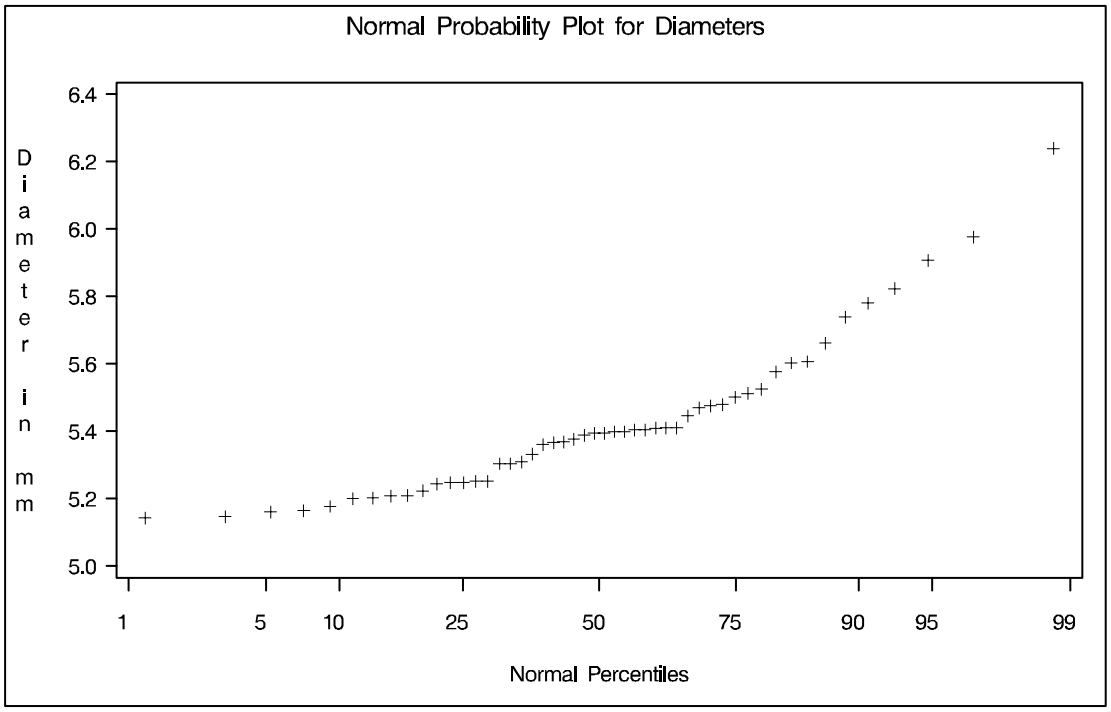


Figure 16.1. Normal Probability Plot Created with Graphics Device

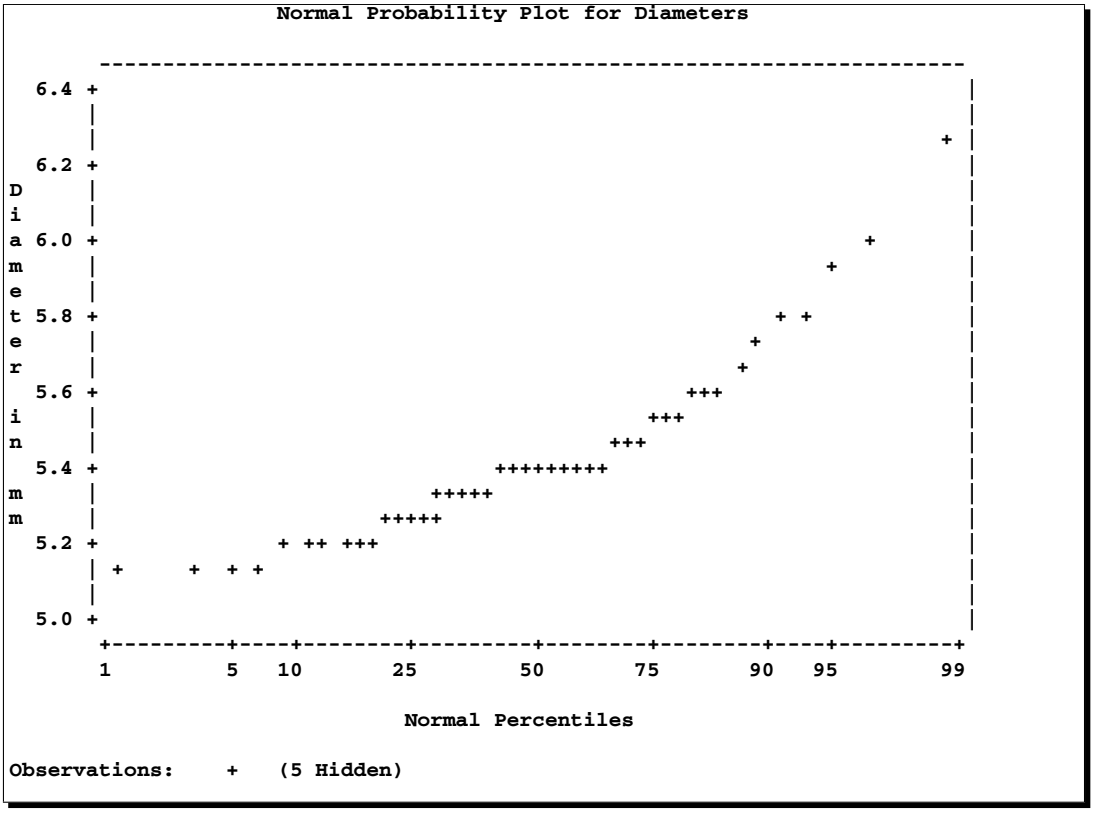


Figure 16.2. Normal Probability Plot Created with Line Printer

Creating Lognormal Probability Plots

See CAPPROB3 in the SAS/QC Sample Library

When you request a lognormal probability plot, you must specify the shape parameter σ for the lognormal distribution (see Table 16.13 on page 455 for the equation). The value of σ must be positive, and typical values of σ range from 0.1 to 1.0. Alternatively, you can specify that σ is to be estimated from the data.

The following statements illustrate the first approach by creating a series of three lognormal probability plots for the variable DIAMETER introduced in the preceding example:

```
symbol v=plus height=3.5pct;
title 'Lognormal Probability Plot for Diameters';
proc capability data=rods noprint;
   probplot diameter / lognormal(theta=est zeta=est sigma=0.2 0.5 0.8)
                       href = 95
                       lhref=1
                       square;
run;
```

The LOGNORMAL option requests plots based on the lognormal family of distributions, and the SIGMA= option requests plots for σ equal to 0.2, 0.5, and 0.8. These plots are displayed in Figure 16.3, Figure 16.4, and Figure 16.5, respectively. The value $\sigma = 0.5$ in Figure 16.4 produces the most linear pattern.

The SQUARE option displays the probability plot in a square format, the HREF= option requests a reference line at the 95^{th} percentile, and the LHREF= option specifies the line type for the reference line.

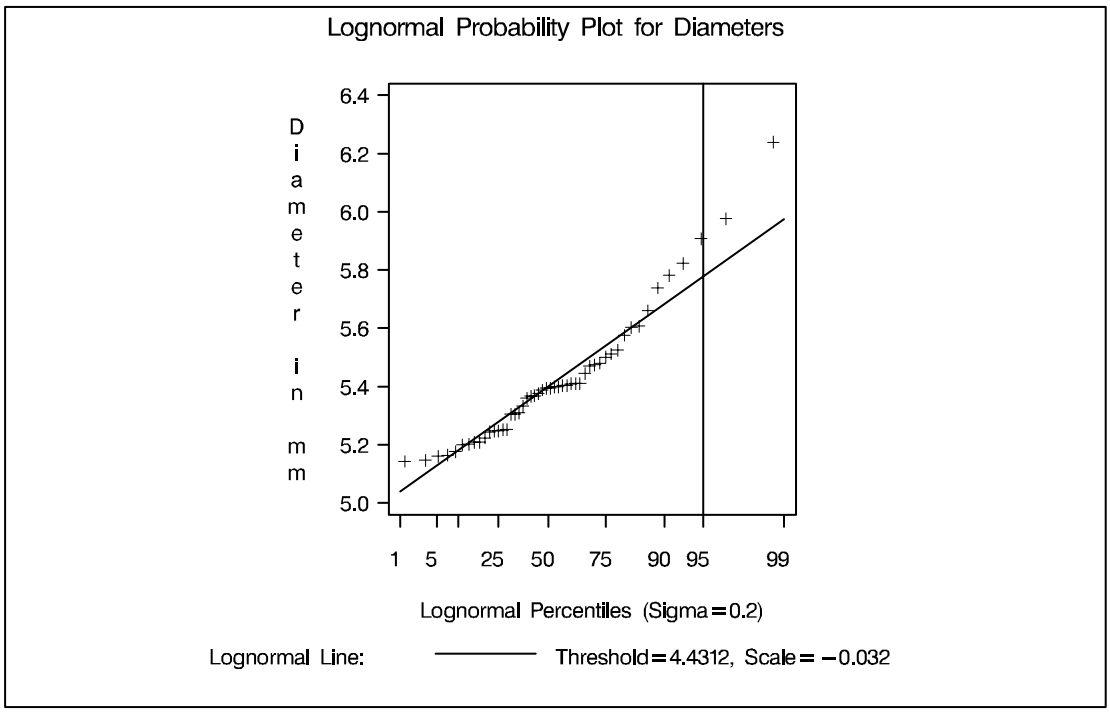


Figure 16.3. Probability Plot Based on Lognormal Distribution with $\sigma = 0.2$

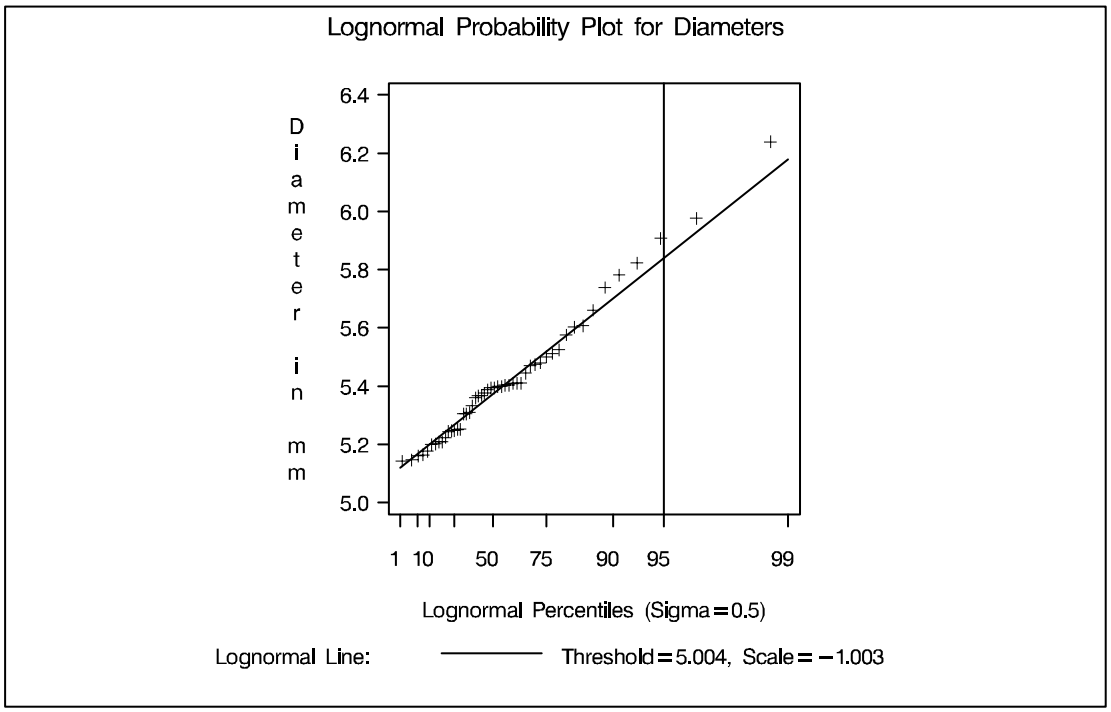


Figure 16.4. Probability Plot Based on Lognormal Distribution with $\sigma = 0.5$

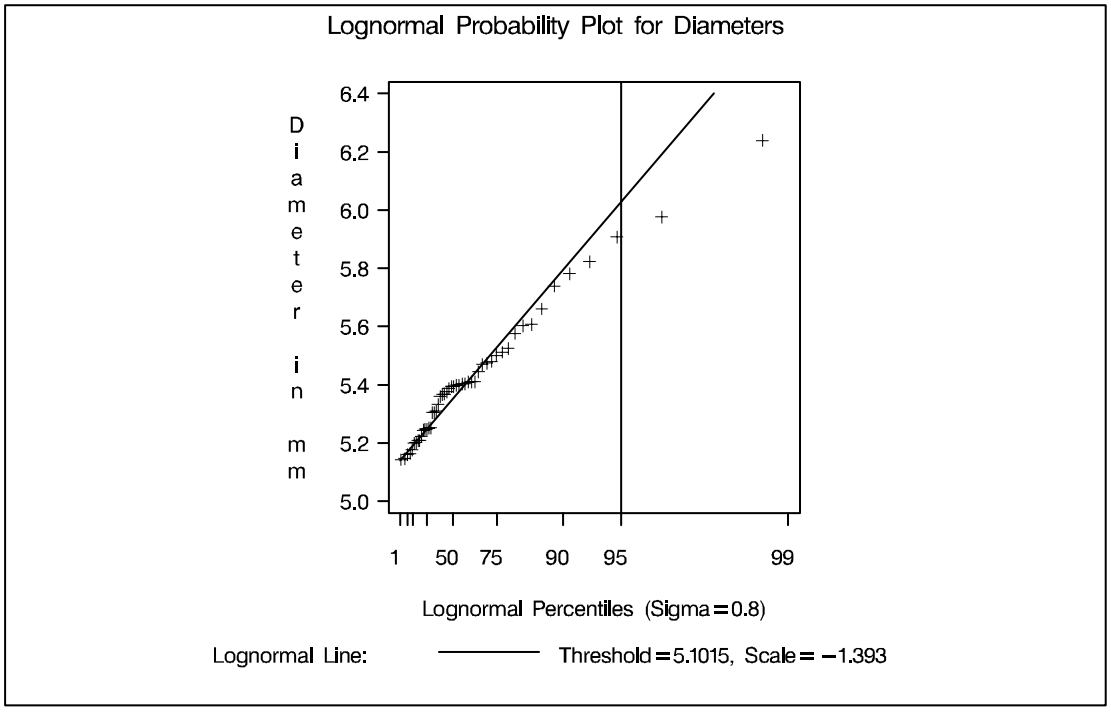


Figure 16.5. Probability Plot Based on Lognormal Distribution with $\sigma = 0.8$

Based on Figure 16.4, the 95^{th} percentile of the diameter distribution is approximately 5.9 mm, since this is the value corresponding to the intersection of the point pattern with the reference line.

The following statements illustrate how you can create a lognormal probability plot for DIAMETER using a local maximum likelihood estimate for σ.

```
symbol v=plus height=3.5pct;
title 'Lognormal Probability Plot for Diameters';
proc capability data=rods noprint;
   probplot diameter / lognormal(theta=est zeta=est sigma=est )
                       href   = 95
                       lhref  = 1
                       square;
run;
```

The plot is displayed in Figure 16.6. Note that the maximum likelihood estimate of σ (in this case 0.041) does not necessarily produce the most linear point pattern. This example is continued in Example 16.2 on page 459.

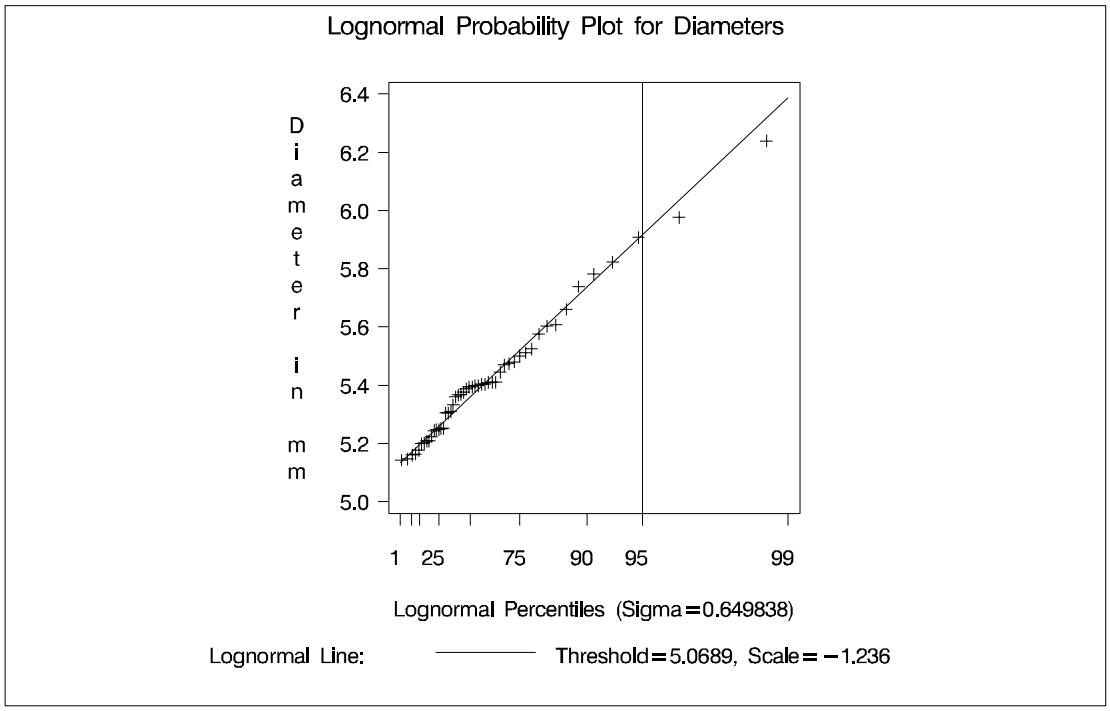


Figure 16.6. Probability Plot Based on Lognormal Distribution with Estimated σ

Syntax

The syntax for the PROBPLOT statement is as follows:

PROBPLOT<*variables* > < **/** *options* >**;**

You can specify the keyword PROB as an alias for PROBPLOT, and you can use any number of PROBPLOT statements in the CAPABILITY procedure. The components of the PROBPLOT statement are described as follows.

variables
are the process variables for which to create probability plots. If you specify a VAR statement, the *variables* must also be listed in the VAR statement. Otherwise, the *variables* can be any numeric variables in the input data set. If you do not specify a list of *variables*, then by default the procedure creates a probability plot for each variable listed in the VAR statement, or for each numeric variable in the DATA= data set if you do not specify a VAR statement. For example, each of the following PROBPLOT statements produces two probability plots, one for LENGTH and one for WIDTH:

```
proc capability data=measures;
   var length width;
   probplot;
run;

proc capability data=measures;
   probplot length width;
run;
```

options
specify the theoretical distribution for the plot or add features to the plot. If you specify more than one variable, the *options* apply equally to each variable. Specify all *options* after the slash (/) in the PROBPLOT statement. You can specify only one *option* naming the distribution in each PROBPLOT statement, but you can specify any number of other *options*. The distributions available are the beta, exponential, gamma, lognormal, normal, two-parameter Weibull, and three-parameter Weibull. By default, the procedure produces a plot for the normal distribution.

In the following example, the NORMAL option requests a normal probability plot for each variable, while the MU= and SIGMA= *normal-options* request a distribution reference line corresponding to the normal distribution with $\mu = 10$ and $\sigma = 0.3$. The SQUARE option displays the plot in a square frame, and the CTEXT= option specifies the text color.

```
proc capability data=measures;
   probplot length1 length2 / normal(mu=10 sigma=0.3)
                              square
                              ctext=blue;
run;
```

Summary of Options

The following tables list the PROBPLOT statement *options* by function. For complete descriptions, see the "Dictionary of Options" section on page 441.

Distribution Options

Table 16.1 summarizes the options for requesting a specific theoretical distribution.

Table 16.1. Keywords to Select a Theoretical Distribution

BETA(*beta-options*)	specifies beta probability plot for shape parameters α, β specified with mandatory ALPHA= and BETA= *beta-options*
EXPONENTIAL(*exponential-options*)	specifies exponential probability plot
GAMMA(*gamma-options*)	specifies gamma probability plot for shape parameter α specified with mandatory ALPHA= *gamma-option*
LOGNORMAL(*lognormal-options*)	specifies lognormal probability plot for shape parameter σ specified with mandatory SIGMA= *lognormal-option*
NORMAL(*normal-options*)	specifies normal probability plot
WEIBULL(*Weibull-options*)	specifies three-parameter Weibull probability plot for shape parameter c specified with mandatory C= *Weibull-option*
WEIBULL2(*Weibull2-options*)	specifies two-parameter Weibull probability plot

Table 16.2 through Table 16.9 summarize options that specify distribution parameters and control the display of a distribution reference line. Specify these options in parentheses after the distribution option. For example, the following statements use the NORMAL option to request a normal probability plot with a distribution reference line:

```
proc capability data=measures;
   probplot length / normal(mu=10 sigma=0.3 color=red);
run;
```

The MU= and SIGMA= *normal-options* display a distribution reference line that corresponds to the normal distribution with mean $\mu_0 = 10$ and standard deviation $\sigma_0 = 0.3$, and the COLOR= *normal-option* specifies the color for the line.

Table 16.2. Reference Line Options Available with All Distributions

COLOR=*color*	specifies color of distribution reference line
L=*linetype*	specifies line type of distribution reference line
SYMBOL='*character*'	specifies plotting character for line printer
W=n	specifies width of distribution reference line

Table 16.3. Beta-Options

ALPHA=*value-list*|EST	specifies mandatory shape parameter α
BETA=*value-list*|EST	specifies mandatory shape parameter β
SIGMA=*value*|EST	specifies σ_0 for distribution reference line
THETA=*value*|EST	specifies θ_0 for distribution reference line

Table 16.4. Exponential-Options

SIGMA=*value*|EST	specifies σ_0 for distribution reference line
THETA=*value*|EST	specifies θ_0 for distribution reference line

Table 16.5. Gamma-Options

ALPHA=*value-list*|EST	specifies mandatory shape parameter α
SIGMA=*value*|EST	specifies σ_0 for distribution reference line
THETA=*value*|EST	specifies θ_0 for distribution reference line

Table 16.6. Lognormal-Options

SIGMA=*value-list*|EST	specifies mandatory shape parameter σ
SLOPE=*value*|EST	specifies slope of distribution reference line
THETA=*value*|EST	specifies θ_0 for distribution reference line
ZETA=*value*|EST	specifies ζ_0 for distribution reference line (slope is $\exp(\zeta_0)$)

Table 16.7. Normal-Options

MU=*value*|EST	specifies μ_0 for distribution reference line
SIGMA=*value*|EST	specifies σ_0 for distribution reference line

Table 16.8. Weibull-Options

C=*value-list*|EST	specifies mandatory shape parameter c
SIGMA=*value*|EST	specifies σ_0 for distribution reference line
THETA=*value*|EST	specifies θ_0 for distribution reference line

Table 16.9. Weibull2-Options

C=*value*|EST	specifies c_0 for distribution reference line (slope is $1/c_0$)
SIGMA=*value*|EST	specifies σ_0 for distribution reference line (intercept is $\log(\sigma_0)$)
SLOPE=*value*|EST	specifies slope of distribution reference line
THETA=*value*	specifies known lower threshold θ_0

General Options

Table 16.10 through Table 16.12 list options that control the appearance of the plots.

Table 16.10. General Plot Layout Options

GRID	specifies reference lines perpendicular to the percentile axis at major tick marks
HREF=*value-list*	specifies reference lines perpendicular to the horizontal axis
HREFLABELS= '*label1*' . . . '*labeln*'	specifies line labels for HREF= lines
LEGEND=*name* | NONE	identifies LEGEND statement
NADJ=*value*	adjusts sample size (N) when computing percentiles
NOFRAME	suppresses frame around plotting area
NOLEGEND	suppresses legend
NOLINELEGEND	suppresses distribution reference line information in legend
NOSPECLEGEND	suppresses specifications information in legend
PCTLMINOR	requests minor tick marks for percentile axis
PCTLORDER=*value-list*	specifies tick mark labels for percentile axis
RANKADJ=*value*	adjusts ranks when computing percentiles
ROTATE	switches horizontal and vertical axes
SQUARE	displays plot in square format
VREF=*value-list*	specifies reference lines perpendicular to the vertical axis
VREFLABELS= '*label1*' . . . '*labeln*'	specifies line labels for VREF= lines

Table 16.11. Options to Enhance Plots Produced on Line Printers

GRIDCHAR='*character*'	specifies character for GRID lines
HREFCHAR='*character*'	specifies character for HREF= lines
NOOBSLEGEND	suppresses legend for hidden points
PROBSYMBOL='*character*'	specifies character for plotted points
VREFCHAR='*character*'	specifies character for VREF= lines

Table 16.12. Options to Enhance Plots Produced on Graphics Devices

ANNOTATE=*SAS-data-set*	provides an annotate data set
CAXIS=*color*	specifies color for axis
CFRAME=*color*	specifies color for frame
CHREF=*color*	specifies color for HREF= lines
CTEXT=*color*	specifies color for text
CVREF=*color*	specifies color for VREF= lines
DESCRIPTION='*string*'	specifies description for graphics catalog member
FONT=*font*	specifies software font for text
HAXIS=*name*	identifies AXIS statement for horizontal axis
HMINOR=*n*	specifies number of minor tick marks on horizontal axis
LGRID=*linetype*	specifies line type for GRID lines
LHREF=*linetype*	specifies line type for HREF= lines
LVREF=*linetype*	specifies line type for VREF= lines
NAME='*string*'	specifies name for plot in graphics catalog
VAXIS=*name*	identifies AXIS statement for vertical axis
VMINOR=*value*	specifies number of minor tick marks on vertical axis

Dictionary of Options

The following entries provide detailed descriptions of options for the PROBPLOT statement. The marginal notes *Graphics* and *Line Printer* identify options that apply only to graphics devices and line printers, respectively.

ALPHA=*value-list***|EST**

specifies values for a mandatory shape parameter α ($\alpha > 0$) for probability plots requested with the BETA and GAMMA options. A plot is created for each value specified. For examples, see the entries for the BETA and GAMMA options. If you specify ALPHA=EST, a maximum likelihood estimate is computed for α.

ANNOTATE=*SAS-data-set*
ANNO=*SAS-data-set*

Graphics

specifies an input data set containing annotate variables as described in *SAS/GRAPH Software: Reference*. You can use this data set to add features to the plot. The ANNOTATE= data set specified in the PROBPLOT statement is used for all plots created by the statement. You can also specify an ANNOTATE= data set in the PROC CAPABILITY statement to enhance all plots created by the procedure; for more information, see "ANNOTATE= Data Sets" on page 189.

BETA(ALPHA=*value-list***|EST BETA=***value-list***|EST** <*beta-options* >**)**

creates a beta probability plot for each combination of the shape parameters α and β given by the mandatory ALPHA= and BETA= options. If you specify ALPHA=EST and BETA=EST, a plot is created based on maximum likelihood estimates for α and

β. In the following examples, the first PROBPLOT statement produces one plot, the second statement produces four plots, the third statement produces six plots, and the fourth statement produces one plot:

```
proc capability data=measures;
   probplot width / beta(alpha=2 beta=2);
   probplot width / beta(alpha=2 3 beta=1 2);
   probplot width / beta(alpha=2 to 3 beta=1 to 2 by 0.5);
   probplot width / beta(alpha=est beta=est);
run;
```

To create the plot, the observations are ordered from smallest to largest, and the i^{th} ordered observation is plotted against the quantile $B^{-1}_{\alpha\beta}\left(\frac{i-0.375}{n+0.25}\right)$, where $B^{-1}_{\alpha\beta}(\cdot)$ is the inverse normalized incomplete beta function, n is the number of nonmissing observations, and α and β are the shape parameters of the beta distribution. The horizontal axis is scaled in percentile units.

The point pattern on the plot for ALPHA=α and BETA=β tends to be linear with intercept* θ and slope σ if the data are beta distributed with the specific density function

$$p(x) = \begin{cases} \frac{(x-\theta)^{\alpha-1}(\theta+\sigma-x)^{\beta-1}}{B(\alpha,\beta)\sigma^{(\alpha+\beta-1)}} & \text{for } \theta < x < \theta+\sigma \\ 0 & \text{for } x \le \theta \text{ or } x \ge \theta+\sigma \end{cases}$$

where $B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$ and

$\theta =$ lower threshold parameter
$\sigma =$ scale parameter $(\sigma > 0)$
$\alpha =$ first shape parameter $(\alpha > 0)$
$\beta =$ second shape parameter $(\beta > 0)$

To obtain graphical estimates of α and β, specify lists of values for the ALPHA= and BETA= options, and select the combination of α and β that most nearly linearizes the point pattern.

To assess the point pattern, you can add a diagonal distribution reference line corresponding to θ_0 and σ_0 with the *beta-options* THETA=θ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of θ_0 and σ_0 with the *beta-options* THETA=EST and SIGMA=EST. Specify these options in parentheses, as in the following example:

```
proc capability data=measures;
   probplot width / beta(alpha=2 beta=3 theta=4 sigma=5);
run;
```

Agreement between the reference line and the point pattern indicates that the beta distribution with parameters α, β, θ_0 and σ_0 is a good fit. You can specify the SCALE=

*The intercept and slope are based on the quantile scale for the horizontal axis, which is displayed on a Q-Q plot; see Chapter 17, "QQPLOT Statement."

option as an alias for the SIGMA= option and the THRESHOLD= option as an alias for the THETA= option.

BETA=*value-list***|EST**
specifies values for the shape parameter β ($\beta > 0$) for probability plots requested with the BETA distribution option. A plot is created for each value specified with the BETA= option. If you specify BETA=EST, a maximum likelihood estimate is computed for β. For examples, see the preceding entry for the BETA option.

C=*value(-list)***|EST**
specifies the shape parameter c ($c > 0$) for probability plots requested with the WEIBULL and WEIBULL2 options. You must specify C= as a *Weibull-option* with the WEIBULL option; in this situation it accepts a list of values, or if you specify C=EST, a maximum likelihood estimate is computed for c. You can optionally specify C=*value* or C=EST as a *Weibull2-option* with the WEIBULL2 option to request a distribution reference line; in this situation, you must also specify SIGMA=*value* or SIGMA=EST.

For example, the first PROBPLOT statement below creates three three-parameter Weibull plots corresponding to the shape parameters $c = 1$, $c = 2$, and $c = 3$. The second PROBPLOT statement creates a single three-parameter Weibull plot corresponding to an estimated value of c. The third PROBPLOT statement creates a single two-parameter Weibull plot with a distribution reference line corresponding to $c_0 = 2$ and $\sigma_0 = 3$.

```
proc capability data=measures;
   probplot width / weibull(c=1 2 3);
   probplot width / weibull(c=est);
   probplot width / weibull2(c=2 sigma=3);
run;
```

CAXIS=*color*
CAXES=*color*
specifies the color used for the axes. This option overrides any COLOR= specifications in an AXIS statement. The default is the first color in the device color list. *Graphics*

CFRAME=*color*
CFR=*color*
specifies the fill color for the area enclosed by the axes and frame. This area is not filled by default. *Graphics*

CHREF=*color*
CH=*color*
specifies the color for reference lines requested by the HREF= option. The default is the first color in the device color list. *Graphics*

COLOR=*color*
specifies the color for a diagonal distribution reference line. Specify the COLOR= option in parentheses following a distribution option keyword. The default is the first color in the device color list. *Graphics*

CTEXT=*color*

Graphics

specifies the color for tick mark values and axis labels. The default is the color specified for the CTEXT= option in the most recent GOPTIONS statement.

CVREF=*color*

CV=*color*

Graphics

specifies the color for reference lines requested by the VREF= option. The default is the first color in the device color list.

DESCRIPTION='*string***'**

DES='*string***'**

Graphics

specifies a description, up to 40 characters, that appears in the PROC GREPLAY master menu. The default string is the variable name.

EXPONENTIAL<(*exponential-options***)>**

EXP(<*exponential-options***>)**

creates an exponential probability plot. To create the plot, the observations are ordered from smallest to largest, and the i^{th} ordered observation is plotted against the quantile $-\log\left(1-\frac{i-0.375}{n+0.25}\right)$, where n is the number of nonmissing observations. The horizontal axis is scaled in percentile units.

The point pattern on the plot tends to be linear with intercept* θ and slope σ if the data are exponentially distributed with the specific density function

$$p(x) = \begin{cases} \frac{1}{\sigma}\exp\left(-\frac{x-\theta}{\sigma}\right) & \text{for } x \geq \theta \\ 0 & \text{for } x < \theta \end{cases}$$

where θ is a threshold parameter, and σ is a positive scale parameter.

To assess the point pattern, you can add a diagonal distribution reference line corresponding to θ_0 and σ_0 with the *exponential-options* THETA=θ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of θ_0 and σ_0 with the *exponential-options* THETA=EST and SIGMA=EST. Specify these options in parentheses, as in the following example:

```
proc capability data=measures;
   probplot width / exponential(theta=4 sigma=5);
run;
```

Agreement between the reference line and the point pattern indicates that the exponential distribution with parameters θ_0 and σ_0 is a good fit. You can specify the SCALE= option as an alias for the SIGMA= option and the THRESHOLD= option as an alias for the THETA= option.

FONT=*font*

Graphics

specifies a software font for horizontal and vertical reference line labels and axis labels. You can also specify fonts for axis labels in an AXIS statement. The FONT= font takes precedence over the FTEXT= font you specify in the GOPTIONS statement. Hardware characters are used by default.

*The intercept and slope are based on the quantile scale for the horizontal axis, which is displayed on a Q-Q plot; see Chapter 17, "QQPLOT Statement."

GAMMA(ALPHA=*value-list***|EST** <*gamma-options*> **)**

creates a gamma probability plot for each value of the shape parameter α given by the mandatory ALPHA= option. If you specify ALPHA=EST, a plot is created based on a maximum likelihood estimate for α.

For example, the first PROBPLOT statement below creates three plots corresponding to $\alpha = 0.4$, $\alpha = 0.5$, and $\alpha = 0.6$. The second PROBPLOT statement creates a single plot.

```
proc capability data=measures;
   probplot width / gamma(alpha=0.4 to 0.6 by 0.2);
   probplot width / gamma(alpha=est);
run;
```

To create the plot, the observations are ordered from smallest to largest, and the i^{th} ordered observation is plotted against the quantile $G_{\alpha}^{-1}\left(\frac{i-0.375}{n+0.25}\right)$, where $G_{\alpha}^{-1}(\cdot)$ is the inverse normalized incomplete gamma function, n is the number of nonmissing observations, and α is the shape parameter of the gamma distribution. The horizontal axis is scaled in percentile units.

The point pattern on the plot for ALPHA=α tends to be linear with intercept* θ and slope σ if the data are gamma distributed with the specific density function

$$p(x) = \begin{cases} \frac{1}{\sigma\Gamma(\alpha)}\left(\frac{x-\theta}{\sigma}\right)^{\alpha-1}\exp\left(-\frac{x-\theta}{\sigma}\right) & \text{for } x > \theta \\ 0 & \text{for } x \leq \theta \end{cases}$$

where

$\theta =$ threshold parameter
$\sigma =$ scale parameter $(\sigma > 0)$
$\alpha =$ shape parameter $(\alpha > 0)$

To obtain a graphical estimate of α, specify a list of values for the ALPHA= option, and select the value that most nearly linearizes the point pattern.

To assess the point pattern, you can add a diagonal distribution reference line corresponding to θ_0 and σ_0 with the *gamma-options* THETA=θ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of θ_0 and σ_0 with the *gamma-options* THETA=EST and SIGMA=EST. Specify these options in parentheses, as in the following example:

```
proc capability data=measures;
   probplot width / gamma(alpha=2 theta=3 sigma=4);
run;
```

Agreement between the reference line and the point pattern indicates that the gamma distribution with parameters α, θ_0 and σ_0 is a good fit. You can specify the SCALE=

*The intercept and slope are based on the quantile scale for the horizontal axis, which is displayed on a Q-Q plot; see Chapter 17, "QQPLOT Statement."

option as an alias for the SIGMA= option and the THRESHOLD= option as an alias for the THETA= option.

GRID
draws reference lines perpendicular to the percentile axis at major tick marks.

GRIDCHAR='*character*'
Line Printer
specifies the character used to form the lines requested by the GRID option for a line printer. The default is the vertical bar (|).

HAXIS=*name*
Graphics
specifies the name of an AXIS statement describing the horizontal axis.

HMINOR=*n*
HM=*n*
Graphics
specifies the number of minor tick marks between each major tick mark on the horizontal axis. Minor tick marks are not labeled. The default is 0.

HREF=*value-list*
draws reference lines perpendicular to the horizontal axis at the values specified. For an example, see Output 16.2.1 on page 459.

HREFCHAR='*character*'
Line Printer
specifies the character used to form the reference lines requested by the HREF= option for a line printer. The default is the vertical bar (|).

HREFLABELS='*label1*' . . . '*labeln*'
HREFLABEL='*label1*' . . . '*labeln*'
HREFLAB='*label1*' . . . '*labeln*'
specifies labels for the reference lines requested by the HREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters. For an example, see Output 16.2.1 on page 459.

L=*linetype*
Graphics
specifies the line type for a diagonal distribution reference line. Specify the L= option in parentheses after a distribution option keyword, as illustrated in the entry for the LOGNORMAL option. The default is 1, which produces a solid line.

LEGEND=*name* **| NONE**
specifies the name of a LEGEND statement describing the legend for specification limit reference lines and fitted curves. Specifying LEGEND=NONE is equivalent to specifying the NOLEGEND option.

LGRID=*linetype*
Graphics
specifies the line type for the reference lines requested by the GRID option. The default is 1, which produces solid lines.

LHREF=*linetype*
LH=*linetype*
Graphics
specifies the line type for reference lines requested by the HREF= option. For an example, see Output 16.2.1 on page 459. The default is 2, which produces a dashed line.

LOGNORMAL(SIGMA=*value-list***|EST** <*lognormal-options* >**)**
LNORM(SIGMA=*value-list***|EST** <*lognormal-options* >**)**

creates a lognormal probability plot for each value of the shape parameter σ given by the mandatory SIGMA= option or its alias, the SHAPE= option. If you specify SIGMA=EST, a plot is created based on a maximum likelihood estimate for σ.

For example, the first PROBPLOT statement below produces two plots, and the second PROBPLOT statement produces a single plot:

```
proc capability data=measures;
   probplot width / lognormal(sigma=1.5 2.5 l=2);
   probplot width / lognormal(sigma=est);
run;
```

To create the plot, the observations are ordered from smallest to largest, and the i^{th} ordered observation is plotted against the quantile $\exp\left(\sigma\Phi^{-1}\left(\frac{i-0.375}{n+0.25}\right)\right)$, where $\Phi^{-1}(\cdot)$ is the inverse standard cumulative normal distribution, n is the number of nonmissing observations, and σ is the shape parameter of the lognormal distribution. The horizontal axis is scaled in percentile units.

The point pattern on the plot for SIGMA=σ tends to be linear with intercept* θ and slope $\exp(\zeta)$ if the data are lognormally distributed with the specific density function

$$p(x) = \begin{cases} \frac{1}{\sigma\sqrt{2\pi}(x-\theta)} \exp\left(-\frac{(\log(x-\theta)-\zeta)^2}{2\sigma^2}\right) & \text{for } x > \theta \\ 0 & \text{for } x \leq \theta \end{cases}$$

where

$\theta =$ threshold parameter
$\zeta =$ scale parameter
$\sigma =$ shape parameter $(\sigma > 0)$

To obtain a graphical estimate of σ, specify a list of values for the SIGMA= option, and select the value that most nearly linearizes the point pattern.

To assess the point pattern, you can add a diagonal distribution reference line corresponding to θ_0 and ζ_0 with the *lognormal-options* THETA=θ_0 and ZETA=ζ_0. Alternatively, you can add a line corresponding to estimated values of θ_0 and ζ_0 with the *lognormal-options* THETA=EST and ZETA=EST.

Specify these options in parentheses, as in the following example:

```
proc capability data=measures;
   probplot width / lognormal(sigma=2 theta=3 zeta=0);
run;
```

*The intercept and slope are based on the quantile scale for the horizontal axis, which is displayed on a Q-Q plot; see Chapter 17, "QQPLOT Statement."

Agreement between the reference line and the point pattern indicates that the lognormal distribution with parameters σ, θ_0, and ζ_0 is a good fit. See Example 16.2 on page 459 for an example.

You can specify the THRESHOLD= option as an alias for the THETA= option and the SCALE= option as an alias for the ZETA= option.

LVREF=*linetype*

Graphics

specifies the line type for reference lines requested by the VREF= option. For an example, see Output 16.2.1 on page 459. The default is 2, which produces a dashed line.

MU=*value*|**EST**

specifies the mean μ_0 for a normal probability plot requested with the NORMAL option. The MU= and SIGMA= *normal-options* must be specified together, and they request a distribution reference line as illustrated in Example 16.1 on page 457. Specify MU=EST to request a distribution reference line with μ_0 equal to the sample mean.

NADJ=*value*

specifies the adjustment value added to the sample size in the calculation of theoretical percentiles. The default is $\frac{1}{4}$, as recommended by Blom (1958). Also refer to Chambers and others (1983) for additional information.

NAME='*string***'**

Graphics

specifies a name for the plot, up to eight characters, that appears in the PROC GREPLAY master menu. The default name is 'CAPABILI'.

NOFRAME

suppresses the frame around the area bounded by the axes.

NOLEGEND
LEGEND=NONE

suppresses legends for specification limits, fitted curves, distribution lines, and hidden observations.

NOLINELEGEND
NOLINEL

suppresses the legend for the optional distribution reference line.

NOOBSLEGEND
NOOBSL

Line Printer

suppresses the legend that indicates the number of hidden observations.

NORMAL<**(***normal-options***)**>
NORM<**(***normal-options***)**>

creates a normal probability plot. This is the default if you do not specify a distribution option. To create the plot, the observations are ordered from smallest to largest, and the i^{th} ordered observation is plotted against the quantile $\Phi^{-1}\left(\frac{i-0.375}{n+0.25}\right)$, where $\Phi^{-1}(\cdot)$ is the inverse cumulative standard normal distribution, and n is the number of nonmissing observations. The horizontal axis is scaled in percentile units.

The point pattern on the plot tends to be linear with intercept* μ and slope σ if the data are normally distributed with the specific

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad \text{for all } x$$

where μ is the mean and σ is the standard deviation ($\sigma > 0$).

To assess the point pattern, you can add a diagonal distribution reference line corresponding to μ_0 and σ_0 with the *normal-options* MU=μ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of μ_0 and σ_0 with the *normal-options* THETA=EST and SIGMA=EST; the estimates of μ_0 and $]sigma_0$ are the sample mean and sample standard deviation.

Specify these options in parentheses, as in the following example:

```
proc capability data=measures;
   probplot length / normal(mu=10 sigma=0.3);
   probplot length / normal(mu=est sigma=est);
run;
```

Agreement between the reference line and the point pattern indicates that the normal distribution with parameters μ_0 and σ_0 is a good fit.

NOSPECLEGEND
NOSPECL
suppresses the legend for specification limit reference lines.

PCTLMINOR
requests minor tick marks for the percentile axis. See Output 16.2.1 on page 459 for an example.

PCTLORDER=*value-list*
specifies the tick mark values labeled on the theoretical percentile axis. Since the values are percentiles, the labels must be between 0 and 100, exclusive. The values must be listed in increasing order and must cover the plotted percentile range. Otherwise, a default list is used. For example, consider the following:

```
proc capability data=measures;
   probplot length / pctlorder=1 10 25 50 75 90 99;
run;
```

Note that the ORDER= option in the AXIS statement is not supported by the PROBPLOT statement.

PROBSYMBOL='*character***'**
specifies the character used to mark the points when the plot is produced on a line printer. The default is the plus sign (+). *Line Printer*

*The intercept and slope are based on the quantile scale for the horizontal axis, which is displayed on a Q-Q plot; see Chapter 17, "QQPLOT Statement."

RANKADJ=*value*

specifies the adjustment value added to the ranks in the calculation of theoretical percentiles. The default is $-\frac{3}{8}$, as recommended by Blom (1958). Also refer to Chambers and others (1983) for additional information.

ROTATE

Graphics

switches the horizontal and vertical axes so that the theoretical percentiles are plotted vertically while the data are plotted horizontally. Regardless of whether the plot has been rotated, horizontal axis options (such as HAXIS=) still refer to the horizontal axis, and vertical axis options (such as VAXIS=) still refer to the vertical axis. All other options that depend on axis placement adjust to the rotated axes.

SCALE=*value***|EST**

is an alias for the SIGMA= option with the BETA, EXPONENTIAL, GAMMA, WEIBULL and WEIBULL2 options and for the ZETA= option with the LOGNORMAL option. See the entries for the SIGMA= and ZETA= options.

SHAPE=*value-list***|EST**

is an alias for the ALPHA= option with the GAMMA option, for the SIGMA= option with the LOGNORMAL option, and for the C= option with the WEIBULL and WEIBULL2 options. See the entries for the ALPHA=, C=, and SIGMA= options.

SIGMA=*value-list***|EST**

specifies the value of the parameter σ, where $\sigma > 0$. Alternatively, you can specify SIGMA=EST to request a maximum likelihood estimate for σ_0. The interpretation and use of the SIGMA= option depend on the distribution option with which it is specified, as indicated by the following table:

Distribution Option	Use of the SIGMA= Option
BETA EXPONENTIAL GAMMA WEIBULL	THETA=θ_0 and SIGMA=σ_0 request a distribution reference line corresponding to θ_0 and σ_0.
LOGNORMAL	SIGMA=$\sigma_1 \ldots \sigma_n$ requests n probability plots with shape parameters $\sigma_1 \ldots \sigma_n$. The SIGMA= option must be specified.
NORMAL	MU=μ_0 and SIGMA=σ_0 request a distribution reference line corresponding to μ_0 and σ_0. SIGMA=EST requests a line with σ_0 equal to the sample standard deviation.
WEIBULL2	SIGMA=σ_0 and C=c_0 request a distribution reference line corresponding to σ_0 and c_0.

In the following example, the first PROBPLOT statement requests a normal plot with a distribution reference line corresponding to $\mu_0 = 5$ and $\sigma_0 = 2$, and the second PROBPLOT statement requests a lognormal plot with shape parameter $\sigma = 3$:

```
proc capability data=measures;
   probplot length / normal(mu=5 sigma=2);
   probplot width  / lognormal(sigma=3);
run;
```

SLOPE=*value*|**EST**
specifies the slope* for a distribution reference line requested with the LOGNORMAL and WEIBULL2 options.

When you use the SLOPE= option with the LOGNORMAL option, you must also specify a threshold parameter value θ_0 with the THETA= *lognormal-option* to request the line. The SLOPE= option is an alternative to the ZETA= *lognormal-option* for specifying ζ_0, since the slope is equal to $\exp(\zeta_0)$.

When you use the SLOPE= option with the WEIBULL2 option, you must also specify a scale parameter value σ_0 with the SIGMA= *Weibull2-option* to request the line. The SLOPE= option is an alternative to the C= *Weibull2-option* for specifying c_0, since the slope is equal to $1/c_0$. See "Location and Scale Parameters" on page 456.

For example, the first and second PROBPLOT statements below produce the same set of probability plots as the third and fourth PROBPLOT statements:

```
proc capability data=measures;
   probplot width / lognormal(sigma=2 theta=0 zeta=0);
   probplot width / weibull2(sigma=2 theta=0 c=0.25);
   probplot width / lognormal(sigma=2 theta=0 slope=1);
   probplot width / weibull2(sigma=2 theta=0 slope=4);
run;
```

SQUARE
displays the probability plot in a square frame. For an example, see Output 16.2.1 on page 459. The default is a rectangular frame.

SYMBOL='*character***'**
specifies the character used to display the distribution reference line when the plot is created using a line printer. The default character is the first letter of the distribution option keyword. *Line Printer*

THETA=*value*|**EST**
specifies the lower threshold parameter θ for plots requested with the BETA, EXPONENTIAL, GAMMA, LOGNORMAL, WEIBULL, and WEIBULL2 options. When used with the WEIBULL2 option, the THETA= option specifies the known lower threshold θ_0, for which the default is 0. When used with the other distribution options, the THETA= option specifies θ_0 for a distribution reference line; alternatively in this situation, you can specify THETA=EST to request a maximum likelihood estimate for θ_0. To request the line, you must also specify a scale parameter. See Output 16.2.1 on page 459 for an example of the THETA= option with a lognormal probability plot.

THRESHOLD=*value*
is an alias for the THETA= option.

*The intercept and slope are based on the quantile scale for the horizontal axis, which is displayed on a Q-Q plot; see Chapter 17, "QQPLOT Statement."

VAXIS=*name*

Graphics

specifies the name of an AXIS statement describing the vertical axis, as illustrated by Output 16.1.1 on page 458.

VMINOR=*n*
VM=*n*

Graphics

specifies the number of minor tick marks between each major tick mark on the vertical axis. Minor tick marks are not labeled. The default is 0.

VREF=*value-list*

draws reference lines perpendicular to the vertical axis at the values specified. See Output 16.2.1 on page 459 for an example.

VREFCHAR='*character***'**

Line Printer

specifies the character used to form the lines requested by the VREF= option for a line printer. The default is the hyphen (-).

VREFLABELS='*label1***' . . . '***labeln***'**
VREFLABEL='*label1***' . . . '***labeln***'**
VREFLAB='*label1***' . . . '***labeln***'**

specifies labels for the lines requested by the VREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

W=*n*

Graphics

specifies the width in pixels for a diagonal distribution reference line. Specify the W= option in parentheses after a distribution option keyword. For an example, see the entry for the WEIBULL option. The default is 1.

WEIBULL(C=*value-list***|EST** <*Weibull-options* >**)**
WEIB(C=*value-list* <*Weibull-options* >**)**

creates a three-parameter Weibull probability plot for each value of the shape parameter c given by the mandatory C= option or its alias, the SHAPE= option. If you specify C=EST, a plot is created based on a maximum likelihood estimate for c. In the following example, the first PROBPLOT statement creates four plots, and the second PROBPLOT statement creates a single plot:

```
proc capability data=measures;
   probplot width / weibull(c=1.8 to 2.4 by 0.2 w=2);
   probplot width / weibull(c=est);
run;
```

To create the plot, the observations are ordered from smallest to largest, and the i^{th} ordered observation is plotted against the quantile $\left(-\log\left(1-\frac{i-0.375}{n+0.25}\right)\right)^{\frac{1}{c}}$, where n is the number of nonmissing observations, and c is the Weibull distribution shape parameter. The horizontal axis is scaled in percentile units.

The point pattern on the plot for C=c tends to be linear with intercept* θ and slope σ if the data are Weibull distributed with the specific density function

$$p(x) = \begin{cases} \frac{c}{\sigma}\left(\frac{x-\theta}{\sigma}\right)^{c-1} \exp\left(-\left(\frac{x-\theta}{\sigma}\right)^{c}\right) & \text{for } x > \theta \\ 0 & \text{for } x \leq \theta \end{cases}$$

where

$\theta =$ threshold parameter
$\sigma =$ scale parameter $(\sigma > 0)$
$c =$ shape parameter $(c > 0)$

To obtain a graphical estimate of c, specify a list of values for the C= option, and select the value that most nearly linearizes the point pattern.

To assess the point pattern, you can add a diagonal distribution reference line corresponding to θ_0 and σ_0 with the *Weibull-options* THETA=θ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of θ_0 and σ_0 with the *Weibull-options* THETA=EST and SIGMA=EST. Specify these options in parentheses, as in the following example:

```
proc capability data=measures;
   probplot width / weibull(c=2 theta=3 sigma=4);
run;
```

Agreement between the reference line and the point pattern indicates that the Weibull distribution with parameters c, θ_0, and σ_0 is a good fit. You can specify the SCALE= option as an alias for the SIGMA= option and the THRESHOLD= option as an alias for the THETA= option.

WEIBULL2<**(***Weibull2-options***)**>
W2<**(***Weibull2-options***)**>
creates a two-parameter Weibull probability plot. You should use the WEIBULL2 option when your data have a *known* lower threshold θ_0. You can specify the threshold value θ_0 with the THETA= *Weibull2-option* or its alias, the THRESHOLD= *Weibull2-option*. The default is $\theta_0 = 0$.

To create the plot, the observations are ordered from smallest to largest, and the log of the shifted i^{th} ordered observation $x_{(i)}$, denoted by $\log(x_{(i)} - \theta_0)$, is plotted against the quantile $\log\left(-\log\left(1 - \frac{i-0.375}{n+0.25}\right)\right)$, where n is the number of nonmissing observations. The horizontal axis is scaled in percentile units. Note that the C= shape parameter option is not mandatory with the WEIBULL2 option.

The point pattern on the plot for THETA=θ_0 tends to be linear with intercept $\log(\sigma)$ and slope $\frac{1}{c}$ if the data are Weibull distributed with the specific density function

*The intercept and slope are based on the quantile scale for the horizontal axis, which is displayed on a Q-Q plot; see Chapter 17, "QQPLOT Statement."

$$p(x) = \begin{cases} \frac{c}{\sigma}\left(\frac{x-\theta_0}{\sigma}\right)^{c-1} \exp\left(-\left(\frac{x-\theta_0}{\sigma}\right)^{c}\right) & \text{for } x > \theta_0 \\ 0 & \text{for } x \leq \theta_0 \end{cases}$$

where

$\theta_0 =$ known lower threshold
$\sigma =$ scale parameter $(\sigma > 0)$
$c =$ shape parameter $(c > 0)$

An advantage of the two-parameter Weibull plot over the three-parameter Weibull plot is that the parameters c and σ can be estimated from the slope and intercept of the point pattern. A disadvantage is that the two-parameter Weibull distribution applies only in situations where the threshold parameter is known.

To assess the point pattern, you can add a diagonal distribution reference line corresponding to σ_0 and c_0 with the *Weibull2-options* SIGMA=σ_0 and C=c_0. Alternatively, you can add a distribution reference line corresponding to estimated values of σ_0 and c_0 with the *Weibull2-options* SIGMA=EST and C=EST. Specify these options in parentheses, as in the following example:

```
proc capability data=measures;
   probplot width / weibull2(theta=3 sigma=4 c=2);
run;
```

Agreement between the distribution reference line and the point pattern indicates that the Weibull distribution with parameters c_0, θ_0 and σ_0 is a good fit. You can specify the SCALE= option as an alias for the SIGMA= option and the SHAPE= option as an alias for the C= option.

ZETA=*value***|EST**
specifies a value for the scale parameter ζ for lognormal probability plots requested with the LOGNORMAL option. Specify THETA=θ_0 and ZETA=ζ_0 to request a distribution reference line with intercept θ_0 and slope $\exp(\zeta_0)$. See Output 16.2.1 on page 459 for an example.

Details

This section provides details on the following topics:

- distributions supported by the PROBPLOT statement
- SYMBOL statement options

Summary of Theoretical Distributions

You can use the PROBPLOT statement to request probability plots based on the theoretical distributions summarized in the following table:

Table 16.13. Distributions and Parameters

Distribution	Density Function $p(x)$	Range	Parameters		
			Location	Scale	Shape
Beta	$\frac{(x-\theta)^{\alpha-1}(\theta+\sigma-x)^{\beta-1}}{B(\alpha,\beta)\sigma^{(\alpha+\beta-1)}}$	$\theta < x < \theta+\sigma$	θ	σ	α, β
Exponential	$\frac{1}{\sigma}\exp\left(-\frac{x-\theta}{\sigma}\right)$	$x \geq \theta$	θ	σ	
Gamma	$\frac{1}{\sigma\Gamma(\alpha)}\left(\frac{x-\theta}{\sigma}\right)^{\alpha-1}\exp\left(-\frac{x-\theta}{\sigma}\right)$	$x > \theta$	θ	σ	α
Lognormal (3-parameter)	$\frac{1}{\sigma\sqrt{2\pi}(x-\theta)}\exp\left(-\frac{(\log(x-\theta)-\zeta)^2}{2\sigma^2}\right)$	$x > \theta$	θ	ζ	σ
Normal	$\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$	all x	μ	σ	
Weibull (3-parameter)	$\frac{c}{\sigma}\left(\frac{x-\theta}{\sigma}\right)^{c-1}\exp\left(-\left(\frac{x-\theta}{\sigma}\right)^{c}\right)$	$x > \theta$	θ	σ	c
Weibull (2-parameter)	$\frac{c}{\sigma}\left(\frac{x-\theta_0}{\sigma}\right)^{c-1}\exp\left(-\left(\frac{x-\theta_0}{\sigma}\right)^{c}\right)$	$x > \theta_0$	θ_0 (known)	σ	c

You can request these distributions with the BETA, EXPONENTIAL, GAMMA, LOGNORMAL, NORMAL, WEIBULL, and WEIBULL2 options, respectively. If you do not specify a distribution option, a normal probability plot is created.

Shape Parameters

Some of the distribution options in the PROBPLOT statement require you to specify one or two shape parameters in parentheses after the distribution keyword. These are summarized in Table 16.14.

Table 16.14. Shape Parameter Options for the PROBPLOT Statement

Distribution Keyword	Mandatory Shape Parameter Option	Range
BETA	ALPHA=α, BETA=β	$\alpha > 0$, $\beta > 0$
EXPONENTIAL	None	
GAMMA	ALPHA=α	$\alpha > 0$
LOGNORMAL	SIGMA=σ	$\sigma > 0$
NORMAL	None	
WEIBULL	C=c	$c > 0$
WEIBULL2	None	

You can visually estimate the value of a shape parameter by specifying a list of values for the shape parameter option. The PROBPLOT statement produces a separate plot for each value. You can then use the value of the shape parameter producing the

most nearly linear point pattern. Alternatively, you can request that the plot be created using an estimated shape parameter. For an example, see "Creating Lognormal Probability Plots" on page 434.

Location and Scale Parameters

If you specify the location and scale parameters for a distribution (or if you request estimates for these parameters), a diagonal distribution reference line is displayed on the plot. (An exception is the two-parameter Weibull distribution, for which a line is displayed when you specify or estimate the scale and shape parameters.) Agreement between this line and the point pattern indicates that the distribution with these parameters is a good fit. For illustrations, see Example 16.1 on page 457 and Example 16.2 on page 459.

The following table shows how the specified parameters determine the intercept* and slope of the line:

Table 16.15. Intercept and Slope of Distribution Reference Line

Distribution	Parameters			Linear Pattern	
	Location	Scale	Shape	Intercept	Slope
Beta	θ	σ	α , β	θ	σ
Exponential	θ	σ		θ	σ
Gamma	θ	σ	α	θ	σ
Lognormal	θ	ζ	σ	θ	$\exp(\zeta)$
Normal	μ	σ		μ	σ
Weibull (3-parameter)	θ	σ	c	θ	σ
Weibull (2-parameter)	θ_0 (known)	σ	c	$\log(\sigma)$	$\frac{1}{c}$

For the LOGNORMAL and WEIBULL2 options, you can specify the slope directly with the SLOPE= option. That is, for the LOGNORMAL option, specifying THETA=θ_0 and SLOPE=$\exp(\zeta_0)$ displays the same line as specifying THETA=θ_0 and ZETA=ζ_0. For the WEIBULL2 option, specifying SIGMA=σ_0 and SLOPE=$\frac{1}{c_0}$ displays the same line as specifying SIGMA=σ_0 and C=c_0.

SYMBOL Statement Options

In earlier releases of SAS/QC software, graphical features of lower and upper specification lines and diagonal distribution reference lines were controlled with options in the SYMBOL2, SYMBOL3, and SYMBOL4 statements, respectively. These options are still supported, although they have been superseded by options in the PROBPLOT and SPEC statements. The following table summarizes the two sets of options:

*The intercept and slope are based on the quantile scale for the horizontal axis, which is displayed on a Q-Q plot; see Chapter 17, "QQPLOT Statement."

Table 16.16. SYMBOL Statement Options

Feature	Statement and Options	Alternative Statement and Options
Symbol markers	SYMBOL1 Statement	
character	VALUE=*special-symbol*	
color	COLOR=*color*	
font	FONT=*font*	
height	HEIGHT=*value*	
Lower specification line	SPEC Statement	SYMBOL2 Statement
position	LSL=*value*	
color	CLSL=*color*	COLOR=*color*
line type	LLSL=*linetype*	LINE=*linetype*
width	WLSL=*value*	WIDTH=*value*
Upper specification line	SPEC Statement	SYMBOL3 Statement
position	USL=*value*	
color	CUSL=*color*	COLOR=*color*
line type	LUSL=*linetype*	LINE=*linetype*
width	WUSL=*value*	WIDTH=*value*
Target reference line	SPEC Statement	
position	TARGET=*value*	
color	CTARGET=*color*	
line type	LTARGET=*linetype*	
width	WTARGET=*value*	
Distribution reference line	PROBPLOT Statement	SYMBOL4 Statement
color	COLOR=*color*	COLOR=*color*
line type	LINE=*linetype*	LINE=*linetype*
width	WIDTH=*value*	WIDTH=*value*

For an illustration of these options, see Example 16.1 on page 457.

Examples

This section provides advanced examples of the PROBPLOT statement.

Example 16.1. Displaying a Normal Reference Line

Measurements of the distance between two holes cut into 50 steel sheets are saved as values of the variable DISTANCE in the following data set:

See CAPPROB4 in the SAS/QC Sample Library

```
data sheets;
   input distance @@;
   label distance='Hole Distance in cm';
   datalines;
 9.80 10.20 10.27  9.70  9.76
10.11 10.24 10.20 10.24  9.63
 9.99  9.78 10.10 10.21 10.00
 9.96  9.79 10.08  9.79 10.06
10.10  9.95  9.84 10.11  9.93
10.56 10.47  9.42 10.44 10.16
10.11 10.36  9.94  9.77  9.36
```

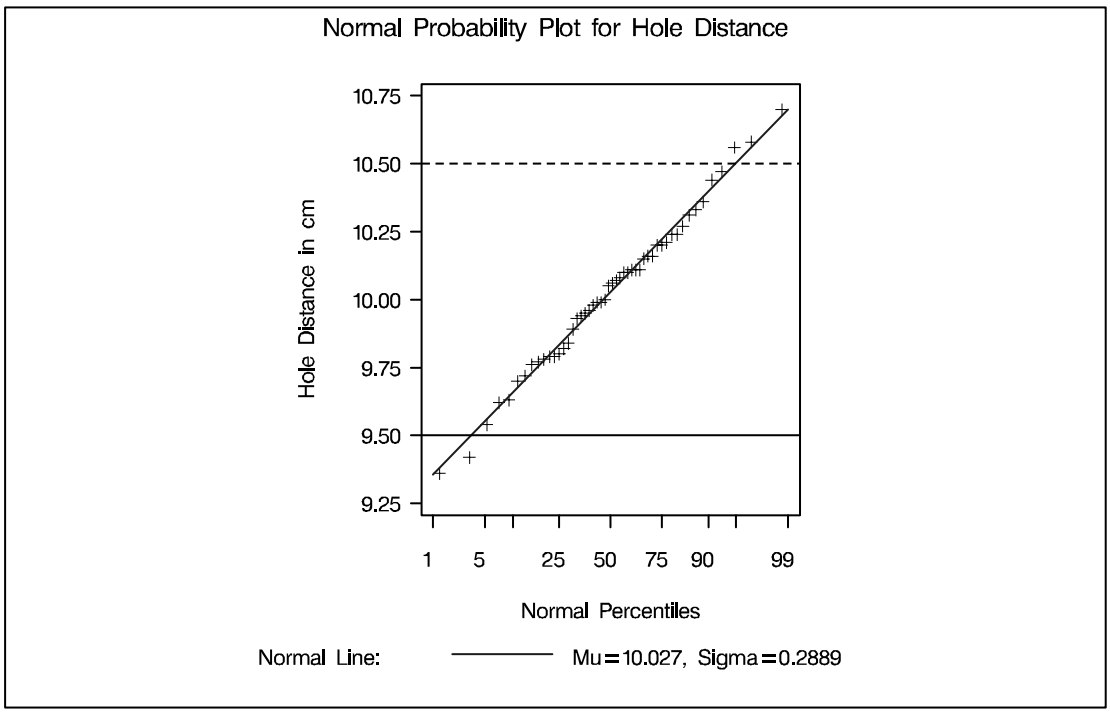
```
 9.89  9.62 10.05  9.72  9.82
 9.99 10.16 10.58 10.70  9.54
10.31 10.07 10.33  9.98 10.15
;
run;
```

The cutting process is in control, and you decide to check whether the process distribution is normal. The following statements create a normal probability plot for DISTANCE with lower and upper specification lines at 9.5 cm and 10.5 cm:

```
symbol v=plus height=3.5pct;
title 'Normal Probability Plot for Hole Distance';
proc capability data=sheets noprint;
   spec  lsl=9.5    llsl=1  clsl=black
         usl=10.5   lusl=2  cusl=black;
   probplot distance / normal(mu=est sigma=est color=blue)
                       square
                       nospeclegend
                       vaxis=axis1;
   axis1 label=(a=90 r=0);
run;
```

The plot is shown in Output 16.1.1. The MU= and SIGMA= *normal-options* request the diagonal reference line that corresponds to the normal distribution with estimated parameters $\hat{\mu} = 10.027$ and $\hat{\sigma} = 0.2889$. The LSL= and USL= SPEC statement options request the lower and upper specification lines, and the LLSL=, LUSL=, CLSL=, and CUSL= options specify the line types and colors. The SYMBOL statement specifies the symbol marker for the plotted points, and the AXIS1 statement specifies the angle and rotation for the vertical axis label.

Output 16.1.1. Normal Reference Line



Example 16.2. Displaying a Lognormal Reference Line

This example is a continuation of "Creating Lognormal Probability Plots" on page 434. Figure 16.4 shows that a lognormal distribution with shape parameter $\sigma = 0.5$ is a good fit for the distribution of DIAMETER in the data set RODS.

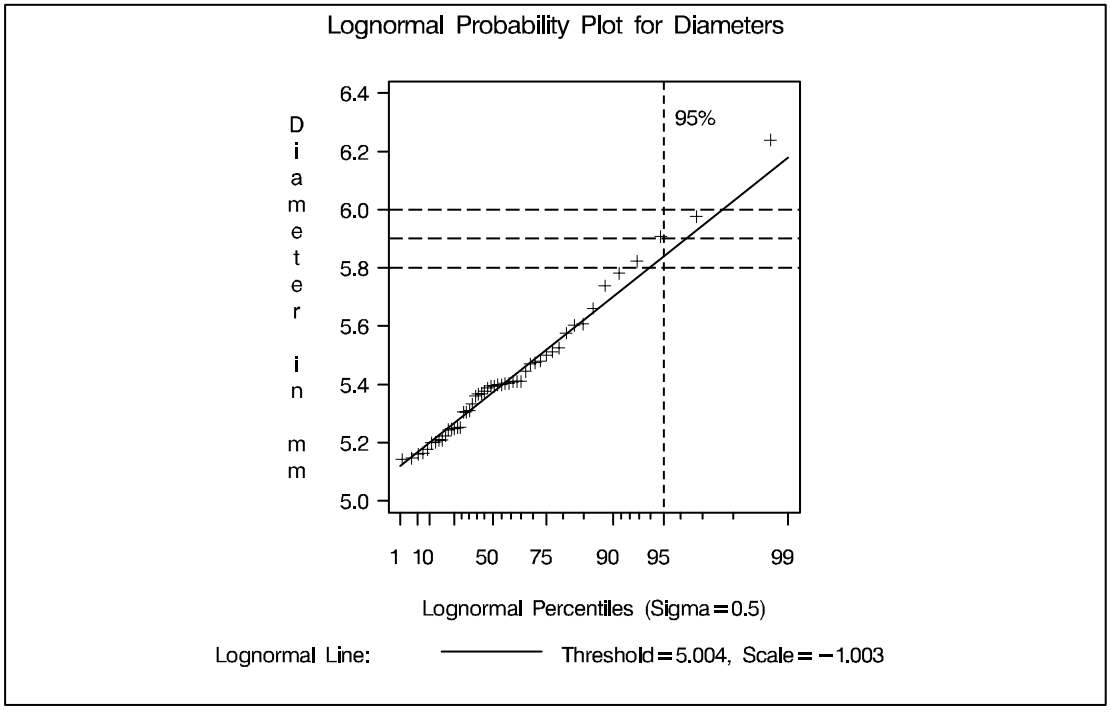
See CAPPROB3 in the SAS/QC Sample Library

The lognormal distribution involves two other parameters: a threshold parameter θ and a scale parameter ζ. See Table 16.13 on page 455 for the equation of the lognormal density function. The following statements illustrate how you can request a diagonal distribution reference line whose slope and intercept are determined by estimates of θ and ζ.

```
symbol v=plus height=3.5pct;
title 'Lognormal Probability Plot for Diameters';
proc capability data=rods noprint;
   probplot diameter / lognormal(sigma=0.5 theta=est zeta=est)
                       square
                       pctlminor
                       href      = 95
                       lhref     = 2
                       hreflabel = '95%'
                       vref      = 5.8 to 6.0 by 0.1
                       lvref     = 3;
run;
```

The plot is shown in Output 16.2.1.

Output 16.2.1. Lognormal Reference Line



The close agreement between the diagonal reference line and the point pattern indicates that the specific lognormal distribution with $\hat{\sigma} = 0.5$, $\hat{\theta} = 5.004$, and $\hat{\zeta} = -1.003$ is a good fit for the diameter measurements.

Specifying HREF=95 adds a reference line indicating the 95^{th} percentile of the lognormal distribution. The LHREF= and HREFLABEL= options specify the line type and a label for this line. The PCTLMINOR option displays minor tick marks on the percentile axis. The VREF= option adds reference lines indicating diameter values of 5.8, 5.9, and 6.0, and the LVREF= option specifies their line type.

Based on the intersection of the diagonal reference line with the HREF= line, the estimated 95^{th} percentile of the diameter distribution is 5.85 mm.

Note that you could also construct a similar plot in which all three parameters are estimated by substituting SIGMA=EST for SIGMA=0.5 in the preceding statements.

Chapter 17
QQPLOT Statement

Chapter Contents

OVERVIEW 463

GETTING STARTED 463
Creating a Normal Quantile-Quantile Plot 464
Adding a Distribution Reference Line 466

SYNTAX 467
Summary of Options 468
Dictionary of Options 471

DETAILS 485
Construction of Quantile-Quantile and Probability Plots 485
Interpretation of Quantile-Quantile and Probability Plots 486
Summary of Theoretical Distributions 487
Graphical Estimation 488
SYMBOL Statement Options 490

EXAMPLES 491
Example 17.1. Interpreting a Normal Q-Q Plot of Nonnormal Data 491
Example 17.2. Estimating Parameters from Lognormal Plots 492
Example 17.3. Comparing Weibull Q-Q Plots 496
Example 17.4. Estimating Cpk from a Normal Q-Q Plot 499

Chapter 17
QQPLOT Statement

Overview

The QQPLOT statement creates a quantile-quantile plot (Q-Q plot), which compares ordered values of a variable with quantiles of a specified theoretical distribution such as the normal. If the data distribution matches the theoretical distribution, the points on the plot form a linear pattern. Thus, you can use a Q-Q plot to determine how well a theoretical distribution models a set of measurements.

You can specify one of the following theoretical distributions with the QQPLOT statement:

- beta
- exponential
- gamma
- three-parameter lognormal
- normal
- two-parameter Weibull
- three-parameter Weibull

You can use options in the QQPLOT statement to

- specify or estimate parameters for the theoretical distribution
- display a reference line corresponding to specific location and scale parameters for the theoretical distribution
- request graphical enhancements

Note: Q-Q plots are similar to probability plots, which you can create with the PROBPLOT statement (see Chapter 16, "PROBPLOT Statement,"). Q-Q plots are preferable for graphical estimation of distribution parameters and capability indices, whereas probability plots are preferable for graphical estimation of percentiles.

Getting Started

The following examples illustrate the basic syntax of the QQPLOT statement. For complete details of the QQPLOT statement, see the "Syntax" section on page 467. Advanced examples are provided on the "Examples" section on page 491.

Creating a Normal Quantile-Quantile Plot

See CAPQQ1 in the SAS/QC Sample Library

Measurements of the distance between two holes cut into 50 steel sheets are saved as values of the variable DISTANCE in the following data set:

```
data sheets;
   input distance @@;
   label distance='Hole Distance in cm';
   datalines;
 9.80 10.20 10.27  9.70  9.76
10.11 10.24 10.20 10.24  9.63
 9.99  9.78 10.10 10.21 10.00
 9.96  9.79 10.08  9.79 10.06
10.10  9.95  9.84 10.11  9.93
10.56 10.47  9.42 10.44 10.16
10.11 10.36  9.94  9.77  9.36
 9.89  9.62 10.05  9.72  9.82
 9.99 10.16 10.58 10.70  9.54
10.31 10.07 10.33  9.98 10.15
;
run;
```

The cutting process is in control, and you decide to check whether the process distribution is normal. The following statements create a Q-Q plot for DISTANCE, shown in Figure 17.1, with lower and upper specification lines at 9.5 cm and 10.5 cm:*

```
symbol v=plus;
title 'Normal Quantile-Quantile Plot for Hole Distance';
proc capability data=sheets noprint;
   spec lsl=9.5 usl=10.5;
   qqplot distance;
run;
```

The plot compares the ordered values of DISTANCE with quantiles of the normal distribution. The linearity of the point pattern indicates that the measurements are normally distributed. Note that a normal Q-Q plot is created by default. If you specify the LINEPRINTER option in the PROC CAPABILITY statement, the plot is created using a line printer, as shown in Figure 17.2. The specification lines are requested with the LSL= and USL= options in the SPEC statement.

*For a P-P plot using these data, see Figure 15.1 on page 411. For a probability plot using these data, see Example 16.2 on page 459.

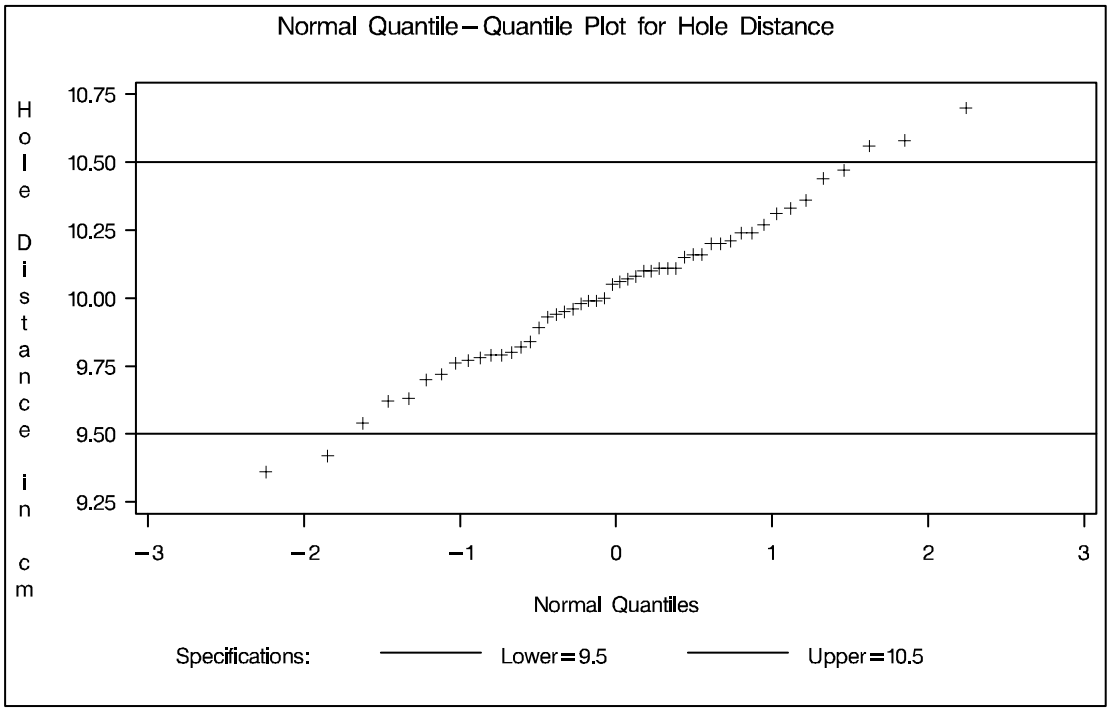


Figure 17.1. Normal Quantile-Quantile Plot Created with Graphics Device

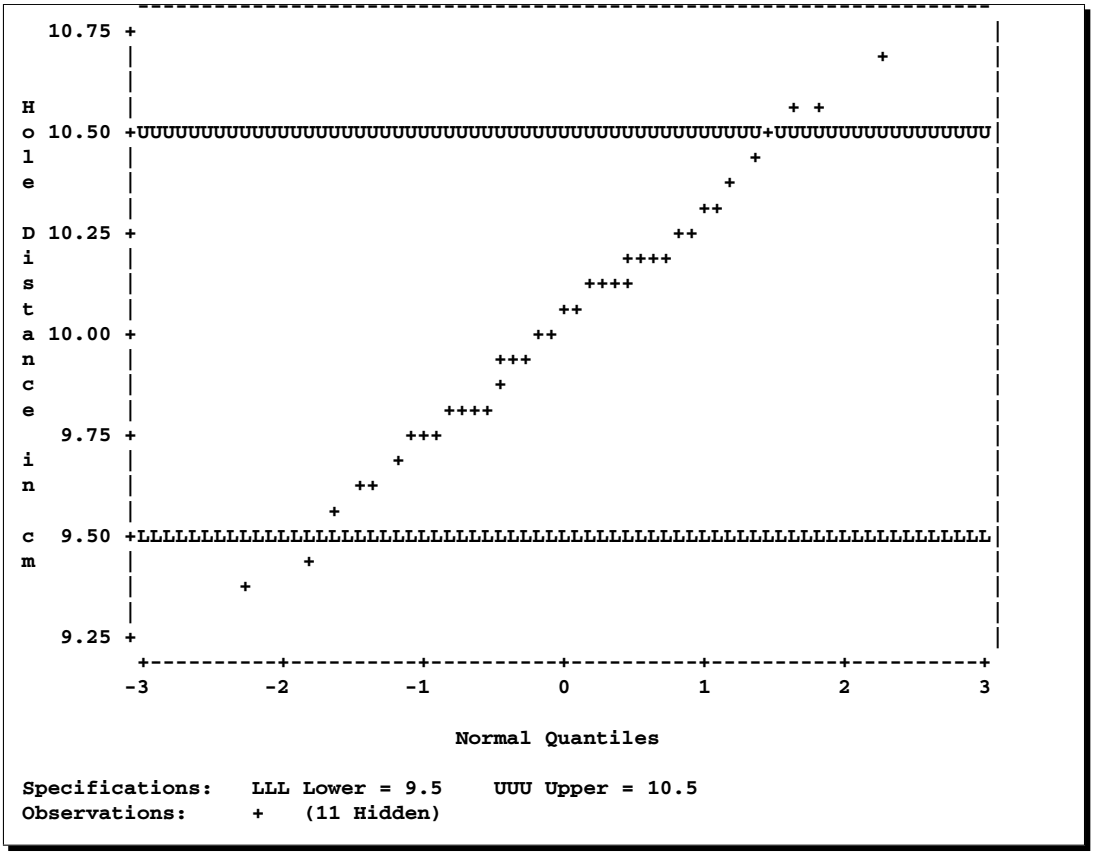


Figure 17.2. Normal Quantile-Quantile Plot Created with Line Printer

Adding a Distribution Reference Line

See CAPQQ1 in the SAS/QC Sample Library

In a normal Q-Q plot, the normal distribution with mean μ_0 and standard deviation σ_0 is represented by a reference line with intercept μ_0 and slope σ_0. The following statements reproduce the Q-Q plot in Figure 17.1, adding the line for which μ_0 and σ_0 are estimated by the sample mean and standard deviation:

```
symbol v=plus;
title 'Normal Quantile-Quantile Plot for Hole Distance';
proc capability data=sheets noprint;
      spec  lsl=9.5   llsl=2   clsl=black
            usl=10.5  lusl=20  cusl=blacl;
   qqplot distance / normal(mu=est sigma=est color=black l=2)
                square
                nospeclegend;
run;
```

The plot is displayed in Figure 17.3.

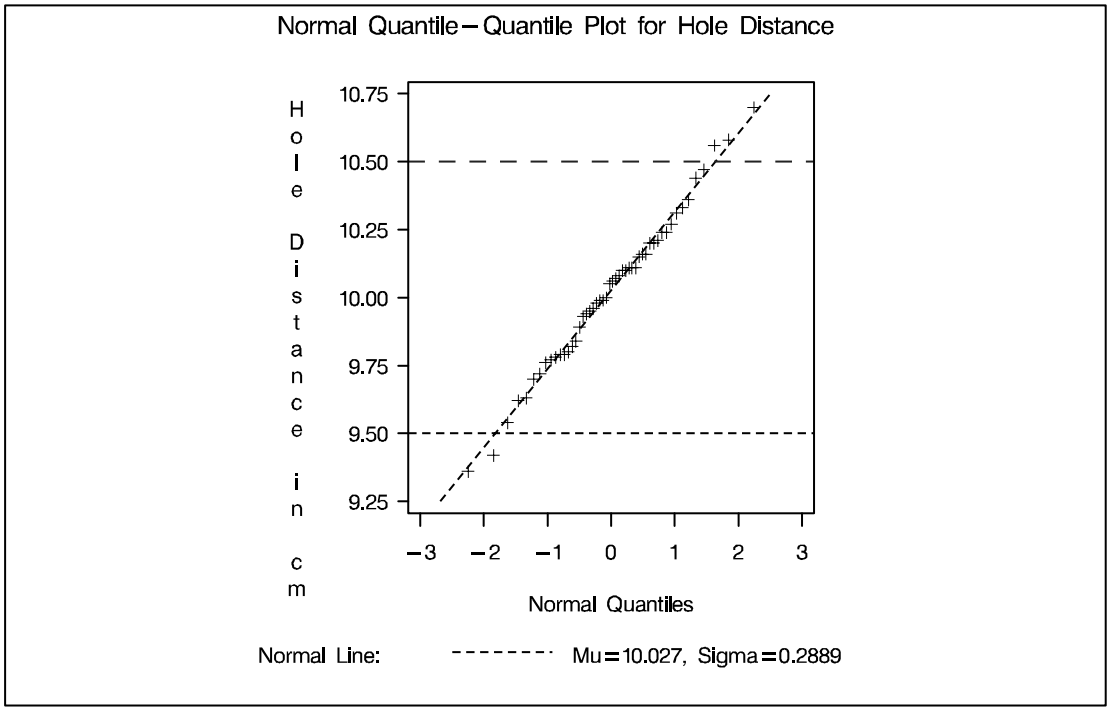


Figure 17.3. Adding a Distribution Reference Line to a Q-Q Plot

Specifying MU=EST and SIGMA=EST with the NORMAL option requests the reference line (alternatively, you can specify numeric values for μ_0 and σ_0 with the MU= and SIGMA= options). The COLOR= and L= options specify the color of the line and the line type. The SQUARE option displays the plot in a square format, and the NOSPECLEGEND option suppresses the legend for the specification lines. The LLSL=, LUSL=, CLSL=, and CUSL= options in the SPEC statement specify line types and colors for the specification limits.

Syntax

The syntax for the QQPLOT statement is as follows:

QQPLOT<*variables* > < **/** *options* >**;**

You can specify the keyword QQ as an alias for QQPLOT, and you can use any number of QQPLOT statements in the CAPABILITY procedure. The components of the QQPLOT statement are described as follows.

variables
are the process variables for which to create Q-Q plots. If you specify a VAR statement, the *variables* must also be listed in the VAR statement. Otherwise, the *variables* can be any numeric variables in the input data set. If you do not specify a list of *variables*, then by default the procedure creates a Q-Q plot for each variable listed in the VAR statement, or for each numeric variable in the DATA= data set if you do not specify a VAR statement. For example, each of the following QQPLOT statements produces two Q-Q plots, one for LENGTH and one for WIDTH:

```
proc capability data=measures;
   var length width;
   qqplot;
run;

proc capability data=measures;
   qqplot length width;
run;
```

options
specify the theoretical distribution for the plot or add features to the plot. If you specify more than one variable, the *options* apply equally to each variable. Specify all *options* after the slash (*/*) in the QQPLOT statement. You can specify only one *option* naming the distribution in each QQPLOT statement, but you can specify any number of other *options*. The distributions available are the beta, exponential, gamma, lognormal, normal, two-parameter Weibull, and three-parameter Weibull. By default, the procedure produces a plot for the normal distribution.

In the following example, the NORMAL option requests a normal Q-Q plot for each variable. The MU= and SIGMA= *normal-options* request a distribution reference line with intercept 10 and slope 0.3 for each plot, corresponding to a normal distribution with mean $\mu = 10$ and standard deviation $\sigma = 0.3$. The SQUARE option displays the plot in a square frame, and the CTEXT= option specifies the text color.

```
proc capability data=measures;
   qqplot length1 length2 / normal(mu=10 sigma=0.3)
                            square
                            ctext=blue;
run;
```

Summary of Options

The following tables list the QQPLOT statement *options* by function. For complete descriptions, see "Dictionary of Options" on page 471.

Distribution Options

Table 17.1 summarizes the options for requesting a specific theoretical distribution.

Table 17.1. Keywords to Select a Theoretical Distribution

BETA(*beta-options*)	specifies beta Q-Q plot for shape parameters α, β specified with mandatory ALPHA= and BETA= *beta-options*
EXPONENTIAL(*exponential-options*)	specifies exponential Q-Q plot
GAMMA(*gamma-options*)	specifies gamma Q-Q plot for shape parameter α specified with mandatory ALPHA= *gamma-option*
LOGNORMAL(*lognormal-options*)	specifies lognormal Q-Q plot for shape parameter σ specified with mandatory SIGMA= *lognormal-option*
NORMAL(*normal-options*)	specifies normal Q-Q plot
WEIBULL(*Weibull-options*)	specifies three-parameter Weibull Q-Q plot for shape parameter c specified with mandatory C= *Weibull-option*
WEIBULL2(*Weibull2-options*)	specifies two-parameter Weibull Q-Q plot

Table 17.2 through Table 17.9 summarize options that specify parameter values for theoretical distributions and that control the display of a distribution reference line. Specify these options in parentheses after the distribution option. For example, the following statements use the NORMAL option to request a normal Q-Q plot with a specific distribution reference line. The MU= and SIGMA= *normal-options* display a distribution reference line with intercept 10 and slope 0.3. The COLOR= *normal-option* draws the line in red.

```
proc capability data=measures;
   qqplot length / normal(mu=10 sigma=0.3 color=red);
run;
```

Table 17.2. Reference Line Options Available with All Distributions

COLOR=*color*	specifies color of distribution reference line
L=*linetype*	specifies line type of distribution reference line
SYMBOL='*character*'	specifies plotting character for line printer
W=n	specifies width of distribution reference line

Table 17.3. Beta-Options

ALPHA=*value-list*|EST	specifies mandatory shape parameter α
BETA=*value-list*|EST	specifies mandatory shape parameter β
SIGMA=*value*|EST	specifies reference line slope σ
THETA=*value*|EST	specifies reference line intercept θ

Table 17.4. Exponential-Options

SIGMA=*value*|EST	specifies reference line slope σ
THETA=*value*|EST	specifies reference line intercept θ

Table 17.5. Gamma-Options

ALPHA=*value-list*|EST	specifies mandatory shape parameter α
SIGMA=*value*|EST	specifies reference line slope σ
THETA=*value*|EST	specifies reference line intercept θ

Table 17.6. Lognormal-Options

SIGMA=*value-list*|EST	specifies mandatory shape parameter σ
SLOPE=*value*|EST	specifies reference line slope
THETA=*value*|EST	specifies reference line intercept θ
ZETA=*value*|EST	specifies reference line slope $\exp(\zeta_0)$

Table 17.7. Normal-Options

CPKREF	specifies vertical reference lines at intersection of specification limits with distribution reference line
CPKSCALE	rescales horizontal axis in C_{pk} units
MU=*value*|EST	specifies reference line intercept μ
SIGMA=*value*|EST	specifies reference line slope σ

Table 17.8. Weibull-Options

C=*value-list*|EST	specifies mandatory shape parameter c
SIGMA=*value*|EST	specifies reference line slope σ
THETA=*value*|EST	specifies reference line intercept θ

Table 17.9. Weibull2-Options

C=*value*|EST	specifies c_0 for reference line (slope is $\frac{1}{c_0}$)
SIGMA=*value*|EST	specifies σ_0 for reference line (intercept is $\log(\sigma_0)$)
SLOPE=*value*|EST	specifies reference line slope
THETA=*value*	specifies known lower threshold θ_0

General Options

Table 17.10 through Table 17.12 list options that control the appearance of the plots.

Table 17.10. General Plot Layout Options

HREF=*value-list*	specifies reference lines perpendicular to the horizontal axis
HREFLABELS= '*label1*' . . . '*labeln*'	specifies labels for HREF= lines
LEGEND=*name* | NONE	specifies LEGEND statement
NADJ=*value*	adjusts sample size (N) when computing quantiles
NOFRAME	suppresses frame around plotting area
NOLEGEND	suppresses legend
NOLINELEGEND	suppresses distribution reference line information in legend
NOSPECLEGEND	suppresses specifications information in legend
PCTLAXIS(*axis-options*)	adds a nonlinear percentile axis
PCTLMINOR	adds minor tick marks to percentile axis
PCTLSCALE	replaces theoretical quantiles with percentiles
RANKADJ=*value*	adjusts ranks when computing quantiles
ROTATE	switches horizontal and vertical axes
SQUARE	displays Q-Q plot in square format
VREF=*value-list*	specifies reference lines perpendicular to the vertical axis
VREFLABELS= '*label1*' . . . '*labeln*'	specifies labels for VREF= lines

Table 17.11. Options to Enhance Plots Produced on Line Printers

HREFCHAR='*character*'	specifies line character for HREF= lines
NOOBSLEGEND	suppresses legend for hidden points
QQSYMBOL='*character*'	specifies character for plotted points
VREFCHAR='*character*'	specifies character for VREF= lines

Table 17.12. Options to Enhance Plots Produced on Graphics Devices

ANNOTATE=*SAS-data-set*	provides an annotate data set
CAXIS=*color*	specifies color for axis
CFRAME=*color*	specifies color for frame
CHREF=*color*	specifies color for HREF= lines
CTEXT=*color*	specifies color for text
CVREF=*color*	specifies color for VREF= lines
DESCRIPTION='*string*'	specifies description for graphics catalog member
FONT=*font*	specifies software font for text
HAXIS=*name*	identifies AXIS statement for horizontal axis
HMINOR=n	specifies number of minor tick marks on horizontal axis
LHREF=*linetype*	specifies line type for HREF= lines
LVREF=*linetype*	specifies line type for VREF= lines
NAME='*string*'	specifies name for plot in graphics catalog
VAXIS=*name*	identifies AXIS statement for vertical axis
VMINOR=*value*	specifies number of minor tick marks on vertical axis

Dictionary of Options

The following entries provide detailed descriptions of options for the QQPLOT statement. The marginal notes *Graphics* and *Line Printer* identify options that apply to graphics devices and line printers, respectively.

ALPHA=*value-list***|EST**

specifies values for a mandatory shape parameter α ($\alpha > 0$) for Q-Q plots requested with the BETA and GAMMA options. A plot is created for each value specified. For examples, see the entries for the BETA and GAMMA options. If you specify ALPHA=EST, a maximum likelihood estimate is computed for α.

ANNOTATE=*SAS-data-set*
ANNO=*SAS-data-set*

Graphics

specifies an input data set containing annotate variables as described in *SAS/GRAPH Software: Reference*. You can use this data set to add features to the plot. The ANNOTATE= data set specified in the QQPLOT statement is used for all plots created by the statement. You can also specify an ANNOTATE= data set in the PROC CAPABILITY statement to enhance all plots created by the procedure; for more information, see "ANNOTATE= Data Sets" on page 189.

BETA(ALPHA=*value-list***|EST BETA=***value-list***|EST** <*beta-options* >**)**

creates a beta Q-Q plot for each combination of the shape parameters α and β given by the mandatory ALPHA= and BETA= options. If you specify ALPHA=EST and BETA=EST, a plot is created based on maximum likelihood estimates for α and β. In the following example, the first QQPLOT statement produces one plot, the second

statement produces four plots, the third statement produces six plots, and the fourth statement produces one plot:

```
proc capability data=measures;
   qqplot width / beta(alpha=2 beta=2);
   qqplot width / beta(alpha=2 3 beta=1 2);
   qqplot width / beta(alpha=2 to 3 beta=1 to 2 by 0.5);
   qqplot width / beta(alpha=est beta=est);
run;
```

To create the plot, the observations are ordered from smallest to largest, and the i^{th} ordered observation is plotted against the quantile $B^{-1}_{\alpha\beta}\left(\frac{i-0.375}{n+0.25}\right)$, where $B^{-1}_{\alpha\beta}(\cdot)$ is the inverse normalized incomplete beta function, n is the number of nonmissing observations, and α and β are the shape parameters of the beta distribution.

The point pattern on the plot for ALPHA=α and BETA=β tends to be linear with intercept θ and slope σ if the data are beta distributed with the specific density function

$$p(x) = \begin{cases} \frac{(x-\theta)^{\alpha-1}(\theta+\sigma-x)^{\beta-1}}{B(\alpha,\beta)\sigma^{(\alpha+\beta-1)}} & \text{for } \theta < x < \theta+\sigma \\ 0 & \text{for } x \le \theta \text{ or } x \ge \theta+\sigma \end{cases}$$

where $B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$, and

$\theta =$ lower threshold parameter
$\sigma =$ scale parameter $(\sigma > 0)$
$\alpha =$ first shape parameter $(\alpha > 0)$
$\beta =$ second shape parameter $(\beta > 0)$

To obtain graphical estimates of α and β, specify lists of values for the ALPHA= and BETA= options, and select the combination of α and β that most nearly linearizes the point pattern. To assess the point pattern, you can add a diagonal distribution reference line with intercept θ_0 and slope σ_0 with the *beta-options* THETA=θ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of θ_0 and slope σ_0 with the *beta-options* THETA=EST and SIGMA=EST. Specify these options in parentheses, as in the following example:

```
proc capability data=measures;
   qqplot width / beta(alpha=2 beta=3 theta=4 sigma=5);
run;
```

Agreement between the reference line and the point pattern indicates that the beta distribution with parameters α, β, θ_0, and σ_0 is a good fit. You can specify the SCALE= option as an alias for the SIGMA= option and the THRESHOLD= option as an alias for the THETA= option.

BETA=*value-list***|EST**
specifies values for the shape parameter β $(\beta > 0)$ for Q-Q plots requested with the BETA distribution option. A plot is created for each value specified with the BETA= option. If you specify BETA=EST, a maximum likelihood estimate is computed for β. For examples, see the preceding entry for the BETA distribution option.

C=*value(-list)***|EST**
specifies the shape parameter c ($c > 0$) for Q-Q plots requested with the WEIBULL and WEIBULL2 options. You must specify C= as a *Weibull-option* with the WEIBULL option; in this situation it accepts a list of values, or if you specify C=EST, a maximum likelihood estimate is computed for c. You can optionally specify C=*value* or C=EST as a *Weibull2-option* with the WEIBULL2 option to request a distribution reference line; in this situation, you must also specify SIGMA=*value* or SIGMA=EST. For an example, see Output 17.3.1 on page 498.

CAXIS=*color*
CAXES=*color*
specifies the color for the axes. This option overrides any COLOR= specifications in an AXIS statement. The default is the first color in the device color list. *Graphics*

CFRAME=*color*
CFR=*color*
specifies the color for shading the area enclosed by the axes and frame. This area is not shaded by default. *Graphics*

CHREF=*color*
CH=*color*
specifies the color for reference lines requested with the HREF= option. The default is the first color in the device color list. *Graphics*

COLOR=*color*
specifies the color for a distribution reference line. Specify the COLOR= option in parentheses following a distribution option keyword. For an example, see Figure 17.3 on page 466. The default is the fourth color in the device color list. *Graphics*

CPKREF
draws reference lines extending from the intersections of the specification limits with the distribution reference line to the quantile axis in plots requested with the NORMAL option. Specify CPKREF in parentheses after the NORMAL option. You can use the CPKREF option with the CPKSCALE option for graphical estimation of the capability indices *CPU*, *CPL*, and C_{pk}, as illustrated in Output 17.4.1 on page 500. *Graphics*

CPKSCALE
rescales the quantile axis in C_{pk} units for plots requested with the NORMAL option. Specify CPKSCALE in parentheses after the NORMAL option. You can use the CPKSCALE option with the CPKREF option for graphical estimation of the capability indices *CPU*, *CPL*, and C_{pk}, as illustrated in Output 17.4.1 on page 500.

CTEXT=*color*
specifies the color for tick mark values and axis labels. The default is the color specified for the CTEXT= option in the most recent GOPTIONS statement. In the absence of a GOPTIONS statement, the default color is the first color in the device color list. *Graphics*

CVREF=*color*
CV=*color*

Graphics

specifies the color for reference lines requested by the VREF= option. The default is the first color in the device color list.

DESCRIPTION='*string***'**
DES='*string***'**

Graphics

specifies a description, up to 40 characters, that appears in the PROC GREPLAY master menu. The default string is the variable name.

EXPONENTIAL(<**(***exponential-options***)**>
EXP<**(***exponential-options***)**>**)**

creates an exponential Q-Q plot. To create the plot, the observations are ordered from smallest to largest, and the i^{th} ordered observation is plotted against the quantile $-\log\left(1-\frac{i-0.375}{n+0.25}\right)$, where n is the number of nonmissing observations.

The pattern on the plot tends to be linear with intercept θ and slope σ if the data are exponentially distributed with the specific density function

$$p(x) = \begin{cases} \frac{1}{\sigma}\exp\left(-\frac{x-\theta}{\sigma}\right) & \text{for } x \geq \theta \\ 0 & \text{for } x < \theta \end{cases}$$

where θ is the threshold parameter, and σ is the scale parameter $(\sigma > 0)$.

To assess the point pattern, you can add a diagonal distribution reference line with intercept θ_0 and slope σ_0 with the *exponential-options* THETA=θ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of θ_0 and slope σ_0 with the *exponential-options* THETA=EST and SIGMA=EST. Specify these options in parentheses, as in the following example: as in the following example:

```
proc capability data=measures;
   qqplot width / exponential(theta=4 sigma=5);
run;
```

Agreement between the reference line and the point pattern indicates that the exponential distribution with parameters θ_0 and σ_0 is a good fit. You can specify the SCALE= option as an alias for the SIGMA= option and the THRESHOLD= option as an alias for the THETA= option.

FONT=*font*

Graphics

specifies a software font for horizontal and vertical reference line labels and axis labels. You can also specify fonts for axis labels in an AXIS statement. The FONT= font takes precedence over the FTEXT= font you specify in the GOPTIONS statement. Hardware characters are used by default.

GAMMA(ALPHA=*value-list***|EST** <*gamma-options*> **)**

creates a gamma Q-Q plot for each value of the shape parameter α given by the mandatory ALPHA= option or its alias, the SHAPE= option. The following example produces three probability plots:

```
proc capability data=measures;
   qqplot width / gamma(alpha=0.4 to 0.6 by 0.1);
run;
```

To create the plot, the observations are ordered from smallest to largest, and the i^{th} ordered observation is plotted against the quantile $G_{\alpha}^{-1}\left(\frac{i-0.375}{n+0.25}\right)$, where $G_{\alpha}^{-1}(\cdot)$ is the inverse normalized incomplete gamma function, n is the number of nonmissing observations, and α is the shape parameter of the gamma distribution.

The pattern on the plot for ALPHA=α tends to be linear with intercept θ and slope σ if the data are gamma distributed with the specific density function

$$p(x) = \begin{cases} \frac{1}{\sigma\Gamma(\alpha)}\left(\frac{x-\theta}{\sigma}\right)^{\alpha-1}\exp\left(-\frac{x-\theta}{\sigma}\right) & \text{for } x > \theta \\ 0 & \text{for } x \leq \theta \end{cases}$$

where

$\theta =$ threshold parameter
$\sigma =$ scale parameter $(\sigma > 0)$
$\alpha =$ shape parameter $(\alpha > 0)$

To obtain a graphical estimate of α, specify a list of values for the ALPHA= option, and select the value that most nearly linearizes the point pattern.

To assess the point pattern, you can add a diagonal distribution reference line with intercept θ_0 and slope σ_0 with the *gamma-options* THETA=θ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of θ_0 and σ_0 with the *gamma-options* THETA=EST and SIGMA=EST. Specify these options in parentheses, as in the following example:

```
proc capability data=measures;
   qqplot width / gamma(alpha=2 theta=3 sigma=4);
run;
```

Agreement between the reference line and the point pattern indicates that the gamma distribution with parameters α, θ_0, and σ_0 is a good fit. You can specify the SCALE= option as an alias for the SIGMA= option and the THRESHOLD= option as an alias for the THETA= option.

HAXIS=*name*
specifies the name of an AXIS statement describing the horizontal axis. *Graphics*

HMINOR=*n*
HM=*n*
specifies the number of minor tick marks between each major tick mark on the horizontal axis. Minor tick marks are not labeled. The default is 0. *Graphics*

HREF=*value-list*
draws reference lines perpendicular to the horizontal axis at the values specified. See Example 17.3 on page 496 for illustrations. Related options include the HREFCHAR=, CHREF=, and LHREF= options.

HREFCHAR='*character***'**

Line Printer

specifies the character used to form the reference lines requested by the HREF= option for a line printer. The default is the vertical bar (|).

HREFLABELS='*label1***' . . . '***labeln***'**
HREFLABEL='*label1***' . . . '***labeln***'**
HREFLAB='*label1***' . . . '***labeln***'**

specifies labels for the reference lines requested by the HREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

L=*linetype*

Graphics

specifies the line type for a distribution reference line. Specify the L= option in parentheses following a distribution option keyword. The default is 1, which produces a solid line.

LEGEND=*name* **| NONE**

specifies the name of a LEGEND statement describing the legend for specification limit reference lines and fitted curves. Specifying LEGEND=NONE is equivalent to specifying the NOLEGEND option.

LHREF=*linetype*
LH=*linetype*

Graphics

specifies the line type for reference lines requested by the HREF= option. The default is 2, which produces a dashed line.

LOGNORMAL(SIGMA=*value-list***|EST** <*lognormal-options* >**)**
LNORM(SIGMA=*value-list***|EST** <*lognormal-options* >**)**

creates a lognormal Q-Q plot for each value of the shape parameter σ given by the mandatory SIGMA= option or its alias, the SHAPE= option. For example,

```
proc capability data=measures;
   qqplot width/ lognormal(shape=1.5 2.5);
run;
```

To create the plot, the observations are ordered from smallest to largest, and the i^{th} ordered observation is plotted against the quantile $\exp\left(\sigma\Phi^{-1}\left(\frac{i-0.375}{n+0.25}\right)\right)$, where $\Phi^{-1}(\cdot)$ is the inverse cumulative standard normal distribution, n is the number of nonmissing observations, and σ is the shape parameter of the lognormal distribution.

The pattern on the plot for SIGMA=σ tends to be linear with intercept θ and slope $\exp(\zeta)$ if the data are lognormally distributed with the specific density function

$$p(x) = \begin{cases} \frac{1}{\sigma\sqrt{2\pi}(x-\theta)} \exp\left(-\frac{(\log(x-\theta)-\zeta)^2}{2\sigma^2}\right) & \text{for } x > \theta \\ 0 & \text{for } x \leq \theta \end{cases}$$

where

$\theta =$ threshold parameter
$\zeta =$ scale parameter
$\sigma =$ shape parameter $(\sigma > 0)$

To obtain a graphical estimate of σ, specify a list of values for the SIGMA= option, and select the value that most nearly linearizes the point pattern. For an illustration, see Example 17.2 on page 492.

To assess the point pattern, you can add a diagonal distribution reference line corresponding to the threshold parameter θ_0 and the scale parameter ζ_0 with the *lognormal-options* THETA=θ_0 and ZETA=ζ_0. Alternatively, you can add a line corresponding to estimated values of θ_0 and ζ_0 with the *lognormal-options* THETA=EST and ZETA=EST. This line has intercept θ_0 and slope $\exp(\zeta_0)$. Agreement between the line and the point pattern indicates that the lognormal distribution with parameters σ, θ_0, and ζ_0 is a good fit. See Output 17.2.4 on page 495 for an example. You can specify the THRESHOLD= option as an alias for the THETA= option and the SCALE= option as an alias for the ZETA= option.

You can also display the reference line by specifying THETA=θ_0, and you can specify the slope with the SLOPE= option. For example, the following two QQPLOT statements produce charts with identical reference lines:

```
proc capability data=measures;
   qqplot width / lognormal(sigma=2 theta=3 zeta=1);
   qqplot width / lognormal(sigma=2 theta=3 slope=2.718);
run;
```

LVREF=*linetype*
LV=*linetype*
specifies the line type for reference lines requested by the VREF= option. The default is 2, which produces a dashed line. *Graphics*

MU=*value*|**EST**
specifies a value for the mean μ for a normal Q-Q plot requested with the NORMAL option. Specify MU=μ_0 and SIGMA=σ_0 to request a distribution reference line with intercept μ_0 and slope σ_0. Specify MU=EST to request a distribution reference line with intercept equal to the sample mean, as illustrated in Figure 17.3 on page 466.

NADJ=*value*
specifies the adjustment value added to the sample size in the calculation of theoretical quantiles. The default is $\frac{1}{4}$, as described by Blom (1958). Also refer to Chambers and others (1983) for additional information.

NAME='*string* '
specifies a name for the plot, up to eight characters, that appears in the PROC GREPLAY master menu. The default name is 'CAPABILI'. *Graphics*

NOFRAME
suppresses the frame around the area bounded by the axes.

NOLEGEND
LEGEND=NONE
suppresses legends for specification limits, fitted curves, distribution lines, and hidden observations. For an example, see Output 17.4.1 on page 500.

NOLINELEGEND
NOLINEL
suppresses the legend for the optional distribution reference line.

NOOBSLEGEND
NOOBSL
Line Printer suppresses the legend that indicates the number of hidden observations.

NORMAL<**(***normal-options***)**>
NORM<**(***normal-options***)**>
creates a normal Q-Q plot. This is the default if you do not specify a distribution option. To create the plot, the observations are ordered from smallest to largest, and the i^{th} ordered observation is plotted against the quantile $\Phi^{-1}\left(\frac{i-0.375}{n+0.25}\right)$, where $\Phi^{-1}(\cdot)$ is the inverse cumulative standard normal distribution, and n is the number of nonmissing observations.

The pattern on the plot tends to be linear with intercept μ and slope σ if the data are normally distributed with the specific density function

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad \text{for all } x$$

where μ is the mean, and σ is the standard deviation $(\sigma > 0)$.

To assess the point pattern, you can add a diagonal distribution reference line with intercept μ_0 and slope σ_0 with the *normal-options* MU=μ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of μ_0 and σ_0 with the *normal-options* THETA=EST and SIGMA=EST; the estimates of μ_0 and $]sigma_0$ are the sample mean and sample standard deviation. Specify these options in parentheses, as in the following example:

```
proc capability data=measures;
   qqplot length / normal(mu=10 sigma=0.3);
run;
```

For an example, see "Adding a Distribution Reference Line" on page 466. Agreement between the reference line and the point pattern indicates that the normal distribution with parameters μ_0 and σ_0 is a good fit. You can specify MU=EST and SIGMA=EST to request a distribution reference line with the sample mean and sample standard deviation as the intercept and slope.

Other *normal-options* include CPKREF and CPKSCALE. The CPKREF option draws reference lines extending from the intersections of specification limits with the distribution reference line to the theoretical quantile axis. The CPKSCALE option rescales the theoretical quantile axis in C_{pk} units. You can use the CPKREF option with the CPKSCALE option for graphical estimation of the capability indices *CPU*, *CPL*, and C_{pk}, as illustrated in Output 17.4.1 on page 500.

NOSPECLEGEND
NOSPECL

suppresses the legend for specification limit reference lines. For an example, see Figure 17.3 on page 466.

PCTLAXIS(*axis-options***)**

adds a nonlinear percentile axis along the frame of the Q-Q plot opposite the theoretical quantile axis. The added axis is identical to the axis for probability plots produced with the PROBPLOT statement. When using the PCTLAXIS option, you must specify HREF= values in quantile units, and you cannot use the NOFRAME option. You can specify the following *axis-options*:

GRID	draws vertical grid lines at major percentiles
GRIDCHAR='*character*'	specifies grid line plotting character on line printer
LABEL='*string*'	specifies label for percentile axis
LGRID=*linetype*	specifies line type for grid

For example, the following statements display the plot in Figure 17.4:

See CAPQQ1 in the SAS/QC Sample Library

```
symbol v=plus;
title 'Normal Quantile-Quantile Plot for Hole Distance';
proc capability data=sheets noprint;
   qqplot distance / normal(mu=est sigma=est color=black)
                     nolegend
                     pctlaxis(grid lgrid=35 label='Normal Percentiles');
run;
```

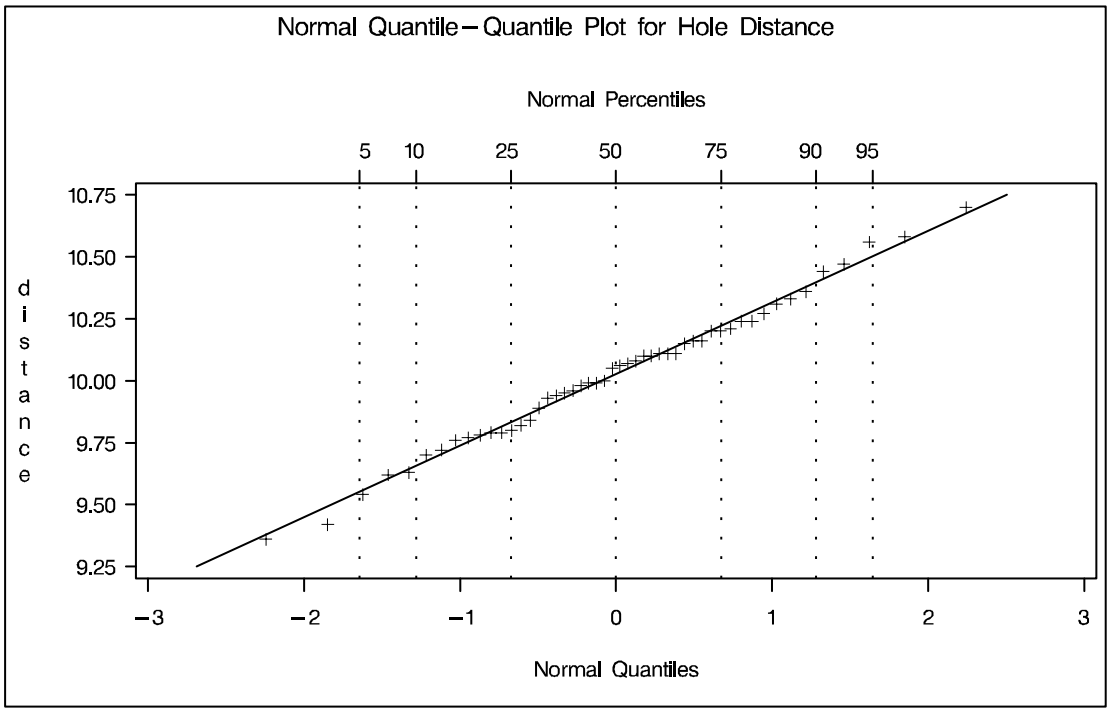


Figure 17.4. Normal Q-Q Plot with Percentile Axis

PCTLMINOR

requests minor tick marks for the percentile axis displayed when you use the PCTLAXIS option. See the entry for the PCTLAXIS option for an example.

PCTLSCALE

requests scale labels for the theoretical quantile axis in percentile units, resulting in a nonlinear axis scale. Tick marks are drawn uniformly across the axis based on the quantile scale. In all other respects, the plot remains the same, and you must specify HREF= values in quantile units. For a true nonlinear axis, use the PCTLAXIS option or use the PROBPLOT statement. For example, the following statements display the plot in Figure 17.5:

See CAPQQ1 in the SAS/QC Sample Library

```
symbol v=plus;
title 'Normal Quantile-Quantile Plot for Hole Distance';
proc capability data=sheets noprint;
   spec  lsl=9.5     usl=10.5
         llsl=2      lusl=20
         clsl=black   cusl=black;
   qqplot distance / normal(mu=est sigma=est cpkref)
                     pctlaxis(grid lgrid=35)
                     nolegend pctlscale;
run;
```

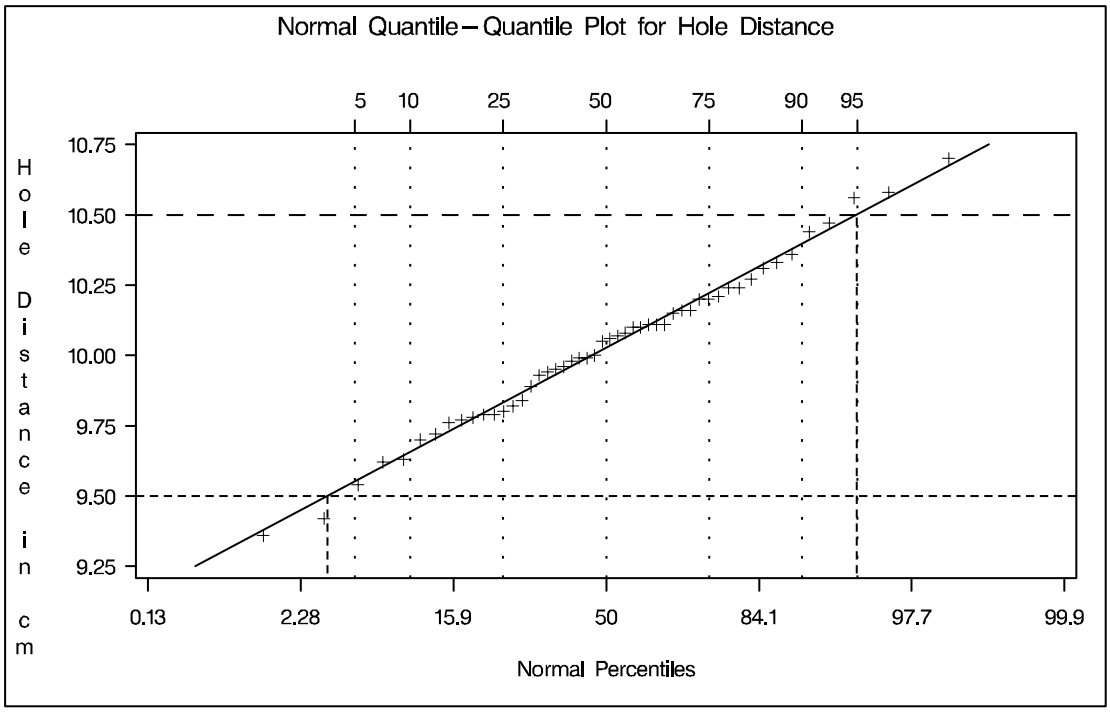


Figure 17.5. Normal Q-Q Plot for Reading Percentiles of Specification Limits

QQSYMBOL='*character***'**

Line Printer

specifies the character used to plot the Q-Q points on a line printer. The default is the plus sign (+).

RANKADJ=*value*

specifies the adjustment value added to the ranks in the calculation of theoretical quantiles. The default is $-\frac{3}{8}$, as described by Blom (1958). Also refer to Chambers and others (1983) for additional information.

ROTATE

Graphics

switches the horizontal and vertical axes so that the theoretical percentiles are plotted vertically while the data are plotted horizontally. Regardless of whether the plot has been rotated, horizontal axis options (such as HAXIS=) refer to the horizontal axis, and vertical axis options (such as VAXIS=) refer to the vertical axis. All other options that depend on axis placement adjust to the rotated axes.

SCALE=*value***|EST**

is an alias for the SIGMA= option with the BETA, EXPONENTIAL, GAMMA, WEIBULL, and WEIBULL2 options and for the ZETA= option with the LOGNORMAL option. See the entries for the SIGMA= and ZETA= options.

SHAPE=*value-list***|EST**

is an alias for the ALPHA= option with the GAMMA option, for the SIGMA= option with the LOGNORMAL option, and for the C= option with the WEIBULL and WEIBULL2 options. See the entries for the ALPHA=, C=, and SIGMA= options.

SIGMA=*value-list***|EST**

specifies the value of the distribution parameter σ, where $\sigma > 0$. Alternatively, you can specify SIGMA=EST to request a maximum likelihood estimate for σ_0. The use of the SIGMA= option depends on the distribution option specified, as indicated by the following table:

Distribution Option	Use of the SIGMA= Option
BETA EXPONENTIAL GAMMA WEIBULL	THETA=θ_0 and SIGMA=σ_0 request a distribution reference line with intercept θ_0 and slope σ_0.
LOGNORMAL	SIGMA=$\sigma_1 \ldots \sigma_n$ requests n Q-Q plots with shape parameters $\sigma_1 \ldots \sigma_n$. The SIGMA= option is mandatory.
NORMAL	MU=μ_0 and SIGMA=σ_0 request a distribution reference line with intercept μ_0 and slope σ_0. SIGMA=EST requests a slope equal to the sample standard deviation.
WEIBULL2	SIGMA=σ_0 and C=c_0 request a distribution reference line with intercept $\log(\sigma_0)$ and slope $\frac{1}{c_0}$.

For an example using SIGMA=EST, see Output 17.4.1 on page 500. For an example of lognormal plots using the SIGMA= option, see Example 17.2 on page 492.

SLOPE=*value***|EST**

specifies the slope for a distribution reference line requested with the LOGNORMAL and WEIBULL2 options.

When you use the SLOPE= option with the LOGNORMAL option, you must also

specify a threshold parameter value θ_0 with the THETA= option. Specifying the SLOPE= option is an alternative to specifying ZETA=ζ_0, which requests a slope of $\exp(\zeta_0)$. See Output 17.2.4 on page 495 for an example.

When you use the SLOPE= option with the WEIBULL2 option, you must also specify a scale parameter value σ_0 with the SIGMA= option. Specifying the SLOPE= option is an alternative to specifying C=c_0, which requests a slope of $\frac{1}{c_0}$.

For example, the first and second QQPLOT statements that follow produce plots identical to those produced by the third and fourth QQPLOT statements:

```
proc capability data=measures;
   qqplot width  / lognormal(sigma=2 theta=0 zeta=0);
   qqplot width  / weibull2(sigma=2 theta=0 c=0.25);
   qqplot width  / lognormal(sigma=2 theta=0 slope=1);
   qqplot width  / weibull2(sigma=2 theta=0 slope=4);
run;
```

For more information, see "Graphical Estimation" on page 488.

SQUARE
displays the Q-Q plot in a square frame. Compare Figure 17.1 on page 465 with Figure 17.3 on page 466. The default is a rectangular frame.

SYMBOL='*character***'**
Line Printer
specifies the character used to plot a distribution reference line when the plot is produced on a line printer. The default character is the first letter of the distribution option keyword.

THETA=*value***|EST**
specifies the lower threshold parameter θ for Q-Q plots requested with the BETA, EXPONENTIAL, GAMMA, LOGNORMAL, WEIBULL, and WEIBULL2 options.

When used with the WEIBULL2 option, the THETA= option specifies the known lower threshold θ_0, for which the default is 0. See Output 17.3.2 on page 499 for an example.

When used with the other distribution options, the THETA= option specifies θ_0 for a distribution reference line; alternatively in this situation, you can specify THETA=EST to request a maximum likelihood estimate for θ_0. To request the line, you must also specify a scale parameter See Output 17.2.4 on page 495 for an example of the THETA= option with a lognormal Q-Q plot.

THRESHOLD=*value***|EST**
is an alias for the THETA= option.

VAXIS=*name*
Graphics
specifies the name of an AXIS statement describing the vertical axis. For an example, see Example 17.1 on page 491.

VMINOR=*n*
VM=*n*
Graphics
specifies the number of minor tick marks between each major tick mark on the vertical axis. Minor tick marks are not labeled. The default is 0.

VREF=*value-list*

draws reference lines perpendicular to the vertical axis at the values specified. For illustrations, see Output 17.2.4 on page 495 or Example 17.3 on page 496. Related options include the VREFCHAR=, CVREF=, and LVREF= options.

VREFCHAR='*character***'**

specifies the character used to form the reference lines requested by the VREF= option for a line printer. The default is the hyphen (-). *Line Printer*

VREFLABELS='*label1***' . . . '***labeln***'**
VREFLABEL='*label1***' . . . '***labeln***'**
VREFLAB='*label1***' . . . '***labeln***'**

specifies labels for the reference lines requested by the VREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

W=*n*

specifies the width in pixels for a distribution reference line, as in the following example. The default is 1. *Graphics*

```
proc capability data=measures;
   qqplot length / normal(mu=5 sigma=2 w=2);
run;
```

WEIBULL(C=*value-list***|EST** <*Weibull-options* >**)**
WEIB(C=*value-list* <*Weibull-options* >**)**

creates a three-parameter Weibull Q-Q plot for each value of the shape parameter c given by the mandatory C= option or its alias, the SHAPE= option. For example,

```
proc capability data=measures;
   qqplot width / weibull(c=1.8 to 2.4 by 0.2);
run;
```

To create the plot, the observations are ordered from smallest to largest, and the i^{th} ordered observation is plotted against the quantile $\left(-\log\left(1-\frac{i-0.375}{n+0.25}\right)\right)^{\frac{1}{c}}$, where n is the number of nonmissing observations, and c is the Weibull distribution shape parameter.

The pattern on the plot for C=c tends to be linear with intercept θ and slope σ if the data are Weibull distributed with the specific density function

$$p(x) = \begin{cases} \frac{c}{\sigma}\left(\frac{x-\theta}{\sigma}\right)^{c-1}\exp\left(-\left(\frac{x-\theta}{\sigma}\right)^{c}\right) & \text{for } x > \theta \\ 0 & \text{for } x \le \theta \end{cases}$$

where θ is the threshold parameter, σ is the scale parameter $(\sigma > 0)$, and c is the shape parameter $(c > 0)$.

To obtain a graphical estimate of c, specify a list of values for the C= option, and select the value that most nearly linearizes the point pattern. For an illustration, see

Example 17.3 on page 496. To assess the point pattern, you can add a diagonal distribution reference line with intercept θ_0 and slope σ_0 with the *Weibull-options* THETA=θ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of θ_0 and σ_0 with the *Weibull-options* THETA=EST and SIGMA=EST. Specify these options in parentheses, as in the following example:

```
proc capability data=measures;
   qqplot width / weibull(c=2 theta=3 sigma=4);
run;
```

Agreement between the reference line and the point pattern indicates that the Weibull distribution with parameters c, θ_0, and σ_0 is a good fit. You can specify the SCALE= option as an alias for the SIGMA= option and the THRESHOLD= option as an alias for the THETA= option.

WEIBULL2<**(***Weibull2-options***)**>
W2<**(***Weibull2-options***)**>
creates a two-parameter Weibull Q-Q plot. You should use the WEIBULL2 option when your data have a *known* lower threshold θ_0. You can specify the threshold value θ_0 with the THETA= option or its alias, the THRESHOLD= option. If you are uncertain of the lower threshold value, you can estimate θ_0 graphically by specifying a list of values for the THETA= option. Select the value that most linearizes the point pattern. The default is $\theta_0 = 0$.

To create the plot, the observations are ordered from smallest to largest, and the log of the shifted i^{th} ordered observation $x_{(i)}$, $\log(x_{(i)} - \theta_0)$, is plotted against the quantile $\log\left(-\log\left(1 - \frac{i-0.375}{n+0.25}\right)\right)$, where n is the number of nonmissing observations. Unlike the three-parameter Weibull quantile, the preceding expression is free of distribution parameters. This is why the C= shape parameter option is not mandatory with the WEIBULL2 option.

The pattern on the plot for THETA=θ_0 tends to be linear with intercept $\log(\sigma)$ and slope $\frac{1}{c}$ if the data are Weibull distributed with the specific density function

$$p(x) = \begin{cases} \frac{c}{\sigma}\left(\frac{x-\theta_0}{\sigma}\right)^{c-1} \exp\left(-\left(\frac{x-\theta_0}{\sigma}\right)^{c}\right) & \text{for } x > \theta_0 \\ 0 & \text{for } x \le \theta_0 \end{cases}$$

where θ_0 is a known lower threshold parameter, σ is a scale parameter $(\sigma > 0)$, and c is a shape parameter $(c > 0)$.

The advantage of a two-parameter Weibull plot over a three-parameter Weibull plot is that you can visually estimate the shape parameter c and the scale parameter σ from the slope and intercept of the point pattern; see Example 17.3 on page 496 for an illustration of this method. The disadvantage is that the two-parameter Weibull distribution applies only in situations where the threshold parameter is known. See "Graphical Estimation" on page 488 for more information.

To assess the point pattern, you can add a diagonal distribution reference line corresponding to the scale parameter σ_0 and shape parameter c_0 with the *Weibull2-options*

SIGMA=σ_0 and C=c_0. Alternatively, you can add a distribution reference line corresponding to estimated values of σ_0 and c_0 with the *Weibull2-options* SIGMA=EST and C=EST. This line has intercept $\log(\sigma_0)$ and slope $\frac{1}{c_0}$. Agreement between the line and the point pattern indicates that the Weibull distribution with parameters c_0, θ_0, and σ_0 is a good fit. You can specify the SCALE= option as an alias for the SIGMA= option and the SHAPE= option as an alias for the C= option.

You can also display the reference line by specifying SIGMA=σ_0, and you can specify the slope with the SLOPE= option. For example, the following QQPLOT statements produce identical plots:

```
proc capability data=measures;
   qqplot width / weibull2(theta=3 sigma=4 c=2);
   qqplot width / weibull2(theta=3 sigma=4 slope=0.5);
run;
```

ZETA=*value***|EST**
specifies a value for the scale parameter ζ for lognormal Q-Q plots requested with the LOGNORMAL option. Specify THETA=θ_0 and ZETA=ζ_0 to request a distribution reference line with intercept θ_0 and slope $\exp(\zeta_0)$.

Details

This section provides details on the following topics:

- construction of Q-Q plots
- interpretation of Q-Q plots
- distributions supported by the QQPLOT statement
- graphical estimation of shape parameters, location and scale parameters, theoretical percentiles, and capability indices
- SYMBOL statement options

Construction of Quantile-Quantile and Probability Plots

Figure 17.6 illustrates how a Q-Q plot is constructed. First, the n nonmissing values of the variable are ordered from smallest to largest:

$$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$$

Then the i^{th} ordered value $x_{(i)}$ is represented on the plot by a point whose y-coordinate is $x_{(i)}$ and whose x-coordinate is $F^{-1}\left(\frac{i-0.375}{n+0.25}\right)$, where $F(\cdot)$ is the theoretical distribution with zero location parameter and unit scale parameter.

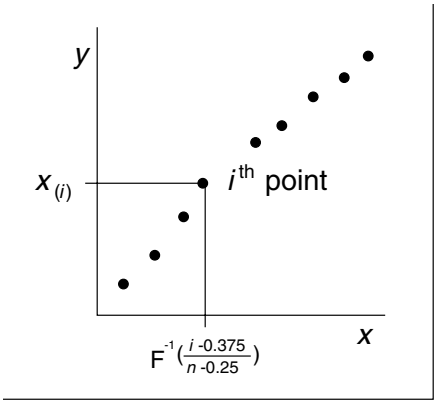


Figure 17.6. Construction of a Q-Q Plot

You can modify the adjustment constants -0.375 and 0.25 with the RANKADJ= and NADJ= options. This default combination is recommended by Blom (1958). For additional information, refer to Chambers and others (1983). Since $x_{(i)}$ is a quantile of the empirical cumulative distribution function (ecdf), a Q-Q plot compares quantiles of the ecdf with quantiles of a theoretical distribution. Probability plots (see Chapter 16, "PROBPLOT Statement,") are constructed the same way, except that the x-axis is scaled nonlinearly in percentiles.

Interpretation of Quantile-Quantile and Probability Plots

The following properties of Q-Q plots and probability plots make them useful diagnostics of how well a specified theoretical distribution fits a set of measurements:

- If the quantiles of the theoretical and data distributions agree, the plotted points fall on or near the line $y = x$.
- If the theoretical and data distributions differ only in their location or scale, the points on the plot fall on or near the line $y = ax + b$. The slope a and intercept b are visual estimates of the scale and location parameters of the theoretical distribution.

Q-Q plots are more convenient than probability plots for graphical estimation of the location and scale parameters since the x-axis of a Q-Q plot is scaled linearly. On the other hand, probability plots are more convenient for estimating percentiles or probabilities.

There are many reasons why the point pattern in a Q-Q plot may not be linear. Chambers and others (1983) and Fowlkes (1987) discuss the interpretations of commonly encountered departures from linearity, and these are summarized in the following table.

Table 17.13. Quantile-Quantile Plot Diagnostics

Description of Point Pattern	Possible Interpretation
All but a few points fall on a line	Outliers in the data
Left end of pattern is below the line; right end of pattern is above the line	Long tails at both ends of the data distribution
Left end of pattern is above the line; right end of pattern is below the line	Short tails at both ends of the data distribution
Curved pattern with slope increasing from left to right	Data distribution is skewed to the right
Curved pattern with slope decreasing from left to right	Data distribution is skewed to the left
Staircase pattern (plateaus and gaps)	Data have been rounded or are discrete

In some applications, a nonlinear pattern may be more revealing than a linear pattern. However, Chambers and others (1983) note that departures from linearity can also be due to chance variation.

Summary of Theoretical Distributions

You can use the QQPLOT statement to request Q-Q plots based on the theoretical distributions summarized in the following table:

Table 17.14. QQPLOT Statement Distribution Options

Distribution	Density Function $p(x)$	Range	Parameters		
			Location	Scale	Shape
Beta	$\frac{(x-\theta)^{\alpha-1}(\theta+\sigma-x)^{\beta-1}}{B(\alpha,\beta)\sigma^{(\alpha+\beta-1)}}$	$\theta < x < \theta+\sigma$	θ	σ	α, β
Exponential	$\frac{1}{\sigma}\exp\left(-\frac{x-\theta}{\sigma}\right)$	$x \geq \theta$	θ	σ	
Gamma	$\frac{1}{\sigma\Gamma(\alpha)}\left(\frac{x-\theta}{\sigma}\right)^{\alpha-1}\exp\left(-\frac{x-\theta}{\sigma}\right)$	$x > \theta$	θ	σ	α
Lognormal (3-parameter)	$\frac{1}{\sigma\sqrt{2\pi}(x-\theta)}\exp\left(-\frac{(\log(x-\theta)-\zeta)^2}{2\sigma^2}\right)$	$x > \theta$	θ	ζ	σ
Normal	$\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$	all x	μ	σ	
Weibull (3-parameter)	$\frac{c}{\sigma}\left(\frac{x-\theta}{\sigma}\right)^{c-1}\exp\left(-\left(\frac{x-\theta}{\sigma}\right)^{c}\right)$	$x > \theta$	θ	σ	c
Weibull (2-parameter)	$\frac{c}{\sigma}\left(\frac{x-\theta_0}{\sigma}\right)^{c-1}\exp\left(-\left(\frac{x-\theta_0}{\sigma}\right)^{c}\right)$	$x > \theta_0$	θ_0 (known)	σ	c

You can request these distributions with the BETA, EXPONENTIAL, GAMMA, LOGNORMAL, NORMAL, WEIBULL, and WEIBULL2 options, respectively. If you do not specify a distribution option, a normal Q-Q plot is created.

Graphical Estimation

You can use Q-Q plots to estimate shape, location, and scale parameters and to estimate percentiles. If you are working with a normal Q-Q plot, you can also estimate certain capability indices.

Shape Parameters

Some distribution options in the QQPLOT statement require that you specify one or two shape parameters in parentheses after the distribution keyword. These are summarized in Table 17.15.

You can visually estimate a shape parameter by specifying a list of values for the shape parameter option. A separate plot is displayed for each value, and you can then select the value that linearizes the point pattern. Alternatively, you can request that the plot be created using an estimated shape parameter. See the entries for the distribution options in "Dictionary of Options" for details on specification of shape parameters. Example 17.2 on page 492 and Example 17.3 on page 496 illustrate shape parameter estimation with lognormal and Weibull Q-Q plots.

Note that for Q-Q plots requested with the WEIBULL2 option, you can estimate the shape parameter c from a linear pattern using the fact that the slope of the pattern is $\frac{1}{c}$. For an illustration, see Example 17.3 on page 496.

Table 17.15. Shape Parameter Options for the QQPLOT Statement

Distribution Keyword	Mandatory Shape Parameter Option	Range
BETA	ALPHA=α, BETA=β	$\alpha > 0, \beta > 0$
EXPONENTIAL	None	
GAMMA	ALPHA=α	$\alpha > 0$
LOGNORMAL	SIGMA=σ	$\sigma > 0$
NORMAL	None	
WEIBULL	C=c	$c > 0$
WEIBULL2	None	

Location and Scale Parameters

When the point pattern on a Q-Q plot is linear, its intercept and slope provide estimates of the location and scale parameters. (An exception to this rule is the two-parameter Weibull distribution, for which the intercept and slope are related to the scale and shape parameters.) Table 17.16 shows how the intercept and slope are related to the parameters for each distribution supported by the QQPLOT statement.

You can enhance a Q-Q plot with a diagonal *distribution reference line* by specifying the parameters that determine the slope and intercept of the line; alternatively, you can request estimates for these parameters. This line is an aid to checking the linearity of the point pattern, and it facilitates parameter estimation. For instance, specifying

Table 17.16. Intercept and Slope of Linear Q-Q Plots

Distribution	Parameters			Linear Pattern	
	Location	Scale	Shape	Intercept	Slope
Beta	θ	σ	α , β	θ	σ
Exponential	θ	σ		θ	σ
Gamma	θ	σ	α	θ	σ
Lognormal	θ	ζ	σ	θ	$\exp(\zeta)$
Normal	μ	σ		μ	σ
Weibull (3-parameter)	θ	σ	c	θ	σ
Weibull (2-parameter)	θ_0 (known)	σ	c	$\log(\sigma)$	$\frac{1}{c}$

MU=3 and SIGMA=2 with the NORMAL option requests a line with intercept 3 and slope 2. Specifying SIGMA=1 and C=2 with the WEIBULL2 option requests a line with intercept $\log(1) = 0$ and slope $\frac{1}{2}$.

With the LOGNORMAL and WEIBULL2 options, you can specify the slope directly with the SLOPE= option. That is, for the LOGNORMAL option, specifying THETA=θ_0 and SLOPE=$\exp(\zeta_0)$ gives the same reference line as specifying THETA=θ_0 and ZETA=ζ_0. For the WEIBULL2 option, specifying SIGMA=σ_0 and SLOPE=$\frac{1}{c_0}$ gives the same reference line as specifying SIGMA=σ_0 and C=c_0.

For an example of parameter estimation using a normal Q-Q plot, see "Adding a Distribution Reference Line" on page 466. Example 17.2 on page 492 illustrates parameter estimation using a lognormal plot, and Example 17.3 on page 496 illustrates estimation using two-parameter and three-parameter Weibull plots.

Theoretical Percentiles

There are two ways to estimate percentiles from a Q-Q plot:

- Specify the PCTLAXIS option, which adds a percentile axis opposite the theoretical quantile axis. The scale for the percentile axis ranges between 0 and 100 with tick marks at percentile values such as 1, 5, 10, 25, 50, 75, 90, 95, and 99. See Figure 17.4 on page 479 for an example.
- Specify the PCTLSCALE option, which relabels the horizontal axis tick marks with their percentile equivalents but does not alter their spacing. For example, on a normal Q-Q plot, the tick mark labeled "0" is relabeled as "50" since the 50^{th} percentile corresponds to the zero quantile. See Figure 17.5 on page 480 for an example.

You can also estimate percentiles using probability plots created with the PROBPLOT statement. See Output 16.2.1 on page 459 for an example.

Capability Indices

When the point pattern on a normal Q-Q plot is linear, you can estimate the capability indices CPU, CPL, and C_{pk} from the plot, as explained by Rodriguez (1992). This method exploits the fact that the horizontal axis of a Q-Q plot indicates the distance in standard deviation units (multiple of σ) between a measurement or specification limit and the process average.

In particular, one-third the standardized distance between an upper specification limit and the mean is the one-sided capability index *CPU*.

$$CPU = \frac{USL - \mu}{3\sigma}$$

Likewise, one-third the standardized distance between a lower specification limit and the mean is the one-sided capability index *CPL*.

$$CPL = \frac{\mu - LSL}{3\sigma}$$

Consequently, if you *rescale* the quantile axis of a normal Q-Q plot by a factor of three, you can read *CPU* and *CPL* from the horizontal coordinates of the points at which the upper and lower specification lines intersect the point pattern. Since C_{pk} is defined as the minimum of *CPU* and *CPL*, this method also provides a graphical estimate of C_{pk}. For an illustration, see Example 17.4 on page 499.

SYMBOL Statement Options

In earlier releases of SAS/QC software, graphical features of lower and upper specification lines and diagonal distribution reference lines were controlled with options in the SYMBOL2, SYMBOL3, and SYMBOL4 statements, respectively. These options are still supported, although they have been superseded by options in the QQPLOT and SPEC statements. The following table summarizes the two sets of options:

Table 17.17. SYMBOL Statement Options

Feature	Statement and Options	Alternative Statement and Options
Symbol markers	SYMBOL1 Statement	
character	VALUE=*special-symbol*	
color	COLOR=*color*	
font	FONT=*font*	
height	HEIGHT=*value*	
Lower specification line	SPEC Statement	SYMBOL2 Statement
position	LSL=*value*	
color	CLSL=*color*	COLOR=*color*
line type	LLSL=*linetype*	LINE=*linetype*
width	WLSL=*value*	WIDTH=*value*
Upper specification line	SPEC Statement	SYMBOL3 Statement
position	USL=*value*	
color	CUSL=*color*	COLOR=*color*
line type	LUSL=*linetype*	LINE=*linetype*
width	WUSL=*value*	WIDTH=*value*
Target reference line	SPEC Statement	
position	TARGET=*value*	
color	CTARGET=*color*	
line type	LTARGET=*linetype*	
width	WTARGET=*value*	
Distribution reference line	QQPLOT Statement	SYMBOL4 Statement
color	COLOR=*color*	COLOR=*color*
line type	LINE=*linetype*	LINE=*linetype*
width	WIDTH=*value*	WIDTH=*value*

Examples

This section provides advanced examples of the QQPLOT statement.

Example 17.1. Interpreting a Normal Q-Q Plot of Nonnormal Data

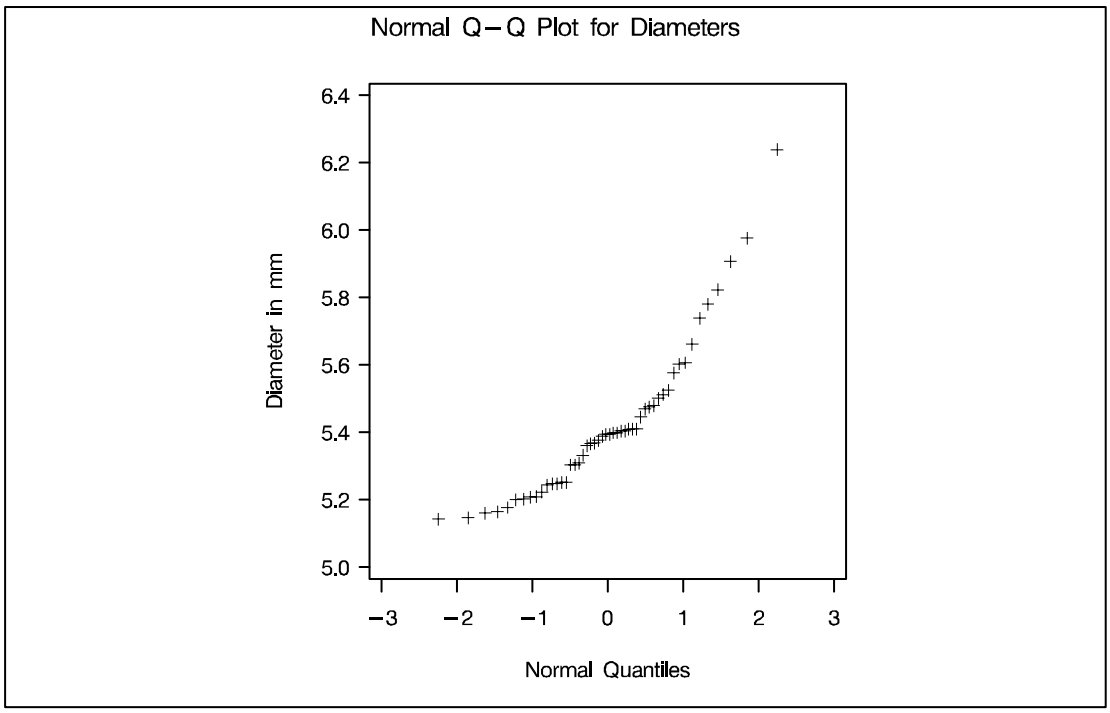
The following statements produce the normal Q-Q plot in Output 17.1.1:

See CAPQQ2 in the SAS/QC Sample Library

```
data measures;
   input diameter @@;
   label diameter='Diameter in mm';
   datalines;
5.501  5.251  5.404  5.366  5.445  5.576  5.607
5.200  5.977  5.177  5.332  5.399  5.661  5.512
5.252  5.404  5.739  5.525  5.160  5.410  5.823
5.376  5.202  5.470  5.410  5.394  5.146  5.244
5.309  5.480  5.388  5.399  5.360  5.368  5.394
5.248  5.409  5.304  6.239  5.781  5.247  5.907
5.208  5.143  5.304  5.603  5.164  5.209  5.475
5.223
 ;
run;
```

```
symbol v=plus;
title 'Normal Q-Q Plot for Diameters';
proc capability data=measures noprint;
   qqplot diameter / normal
                     square
                     vaxis=axis1;
   axis1 label=(a=90 r=0);
run;
```

Output 17.1.1. Normal Quantile-Quantile Plot of Nonnormal Data

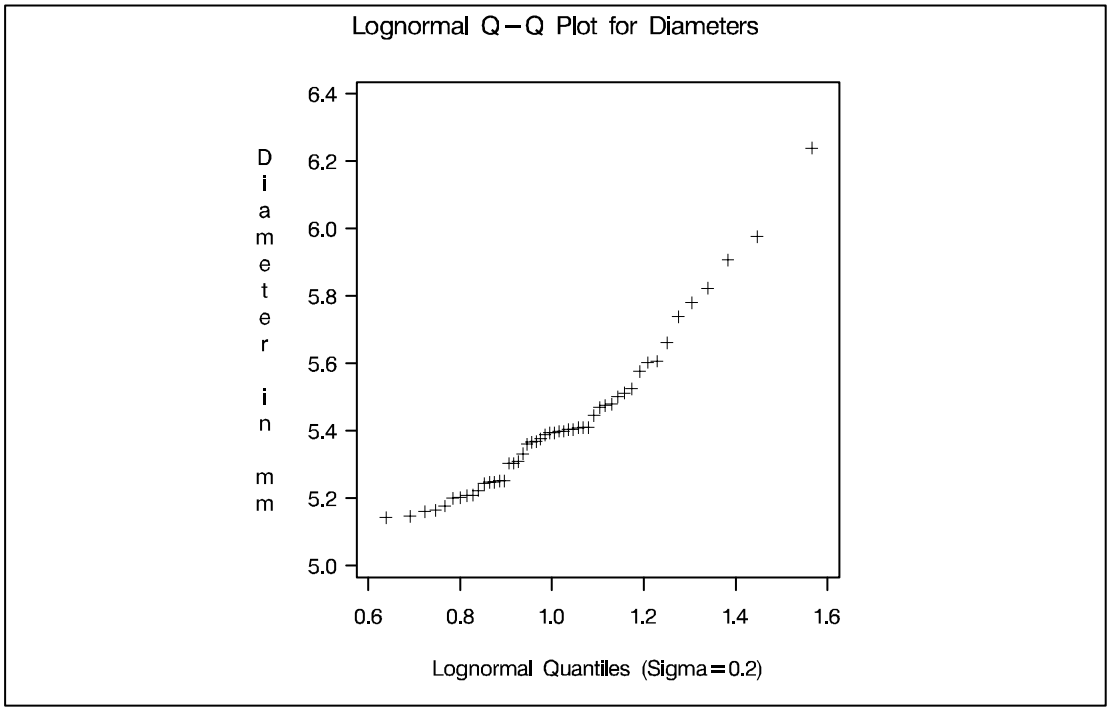


The nonlinearity of the points in Output 17.1.1 indicates a departure from normality. Since the point pattern is curved with slope increasing from left to right, a theoretical distribution that is skewed to the right, such as a lognormal distribution, should provide a better fit than the normal distribution. The mild curvature suggests that you should examine the data with a series of lognormal Q-Q plots for small values of the shape parameter, as illustrated in the next example.

Example 17.2. Estimating Parameters from Lognormal Plots

This example, which is a continuation of Example 17.1, demonstrates techniques for estimating the shape parameter, location and scale parameters, and theoretical percentiles for a lognormal distribution.

Output 17.2.1. Lognormal Quantile-Quantile Plot (σ =0.2)



Three-Parameter Lognormal Plots

See CAPQQ2 in the SAS/QC Sample Library

The three-parameter lognormal distribution depends on a threshold parameter θ, a scale parameter ζ, and a shape parameter σ. You can estimate σ from a series of lognormal Q-Q plots with different values of σ. The estimate is the value of σ that linearizes the point pattern. You can then estimate the threshold and scale parameters from the intercept and slope of the point pattern. The following statements create the series of plots in Output 17.2.1 through Output 17.2.3 for σ values of 0.2, 0.5, and 0.8:

```
symbol v=plus;
title 'Lognormal Q-Q Plot for Diameters';
proc capability data=measures noprint;
   qqplot diameter / lognormal(sigma=0.2 0.5 0.8)
                     square;
run;
```

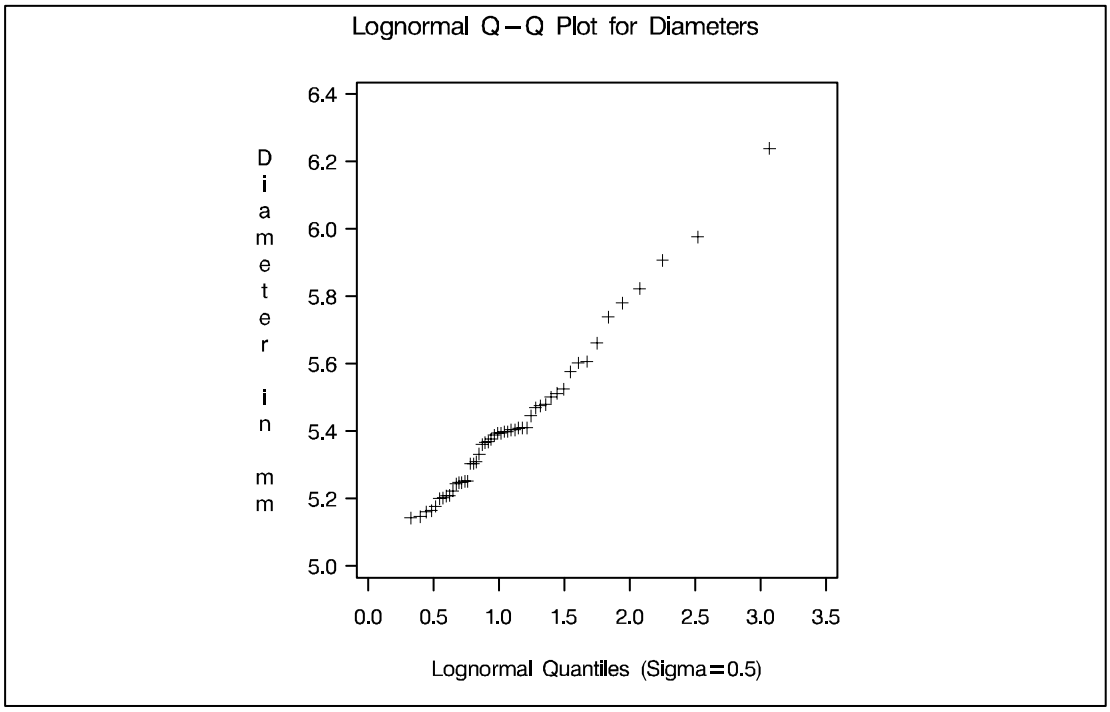
Note: You must specify a value for the shape parameter σ for a lognormal Q-Q plot with the SIGMA= option or its alias, the SHAPE= option.

The plot in Output 17.2.2 displays the most linear point pattern, indicating that the lognormal distribution with $\sigma = 0.5$ provides a reasonable fit for the data distribution.

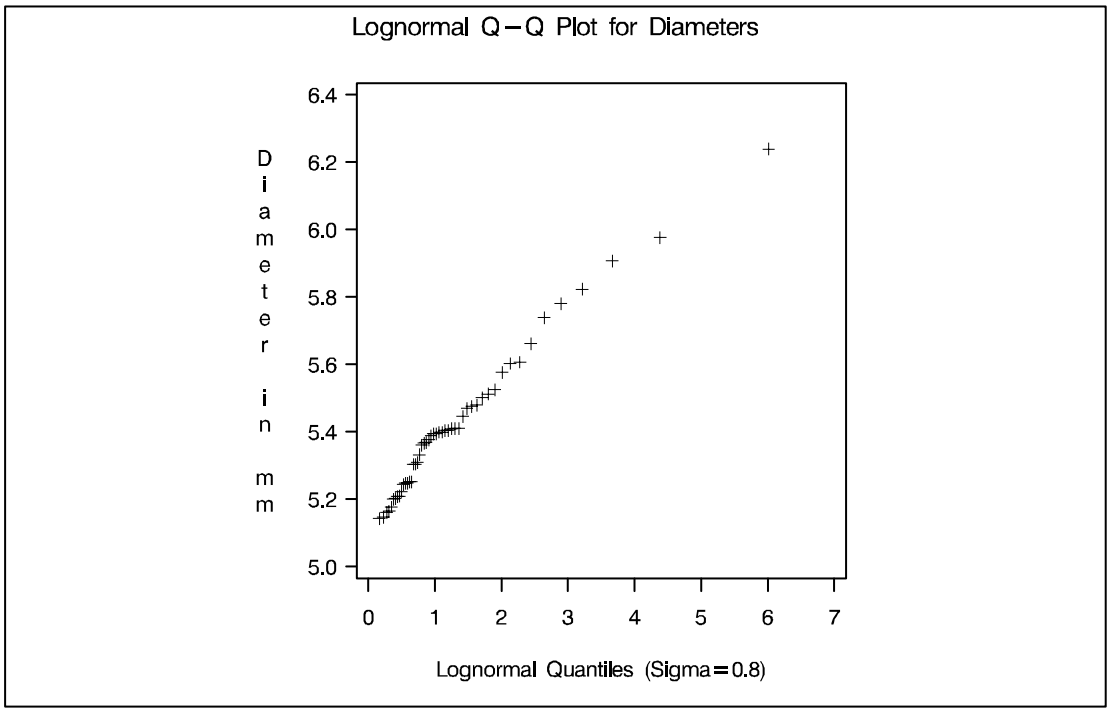
Data with this particular lognormal distribution have the density function

$$p(x) = \begin{cases} \frac{\sqrt{2}}{\sqrt{\pi}(x-\theta)} \exp\left(-2(\log(x-\theta)-\zeta)^2\right) & \text{for } x > \theta \\ 0 & \text{for } x \leq \theta \end{cases}$$

Output 17.2.2. Lognormal Quantile-Quantile Plot (σ =0.5)



Output 17.2.3. Lognormal Quantile-Quantile Plot (σ =0.8)



The points in the plot fall on or near the line with intercept θ and slope $\exp(\zeta)$. Based on Output 17.2.2, $\theta \approx 5$ and $\exp(\zeta) \approx \frac{1.2}{3} = 0.4$, giving $\zeta \approx \log(0.4) \approx -0.92$.

Estimating Percentiles

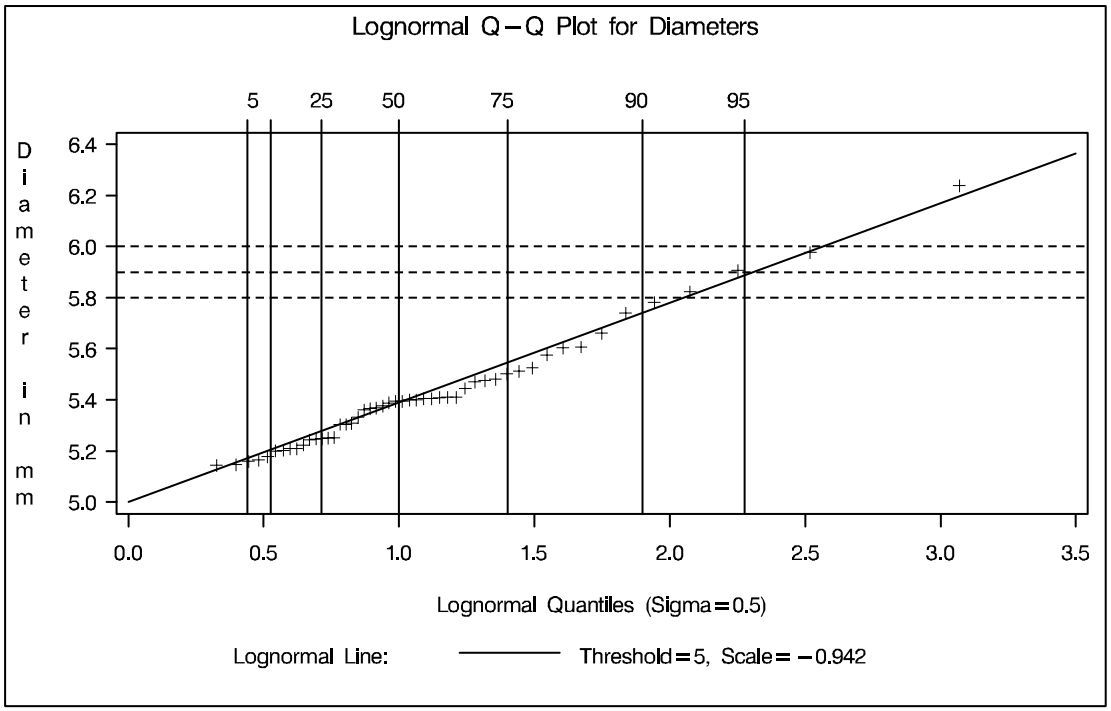
See CAPQQ2 in the SAS/QC Sample Library

You can use a Q-Q plot to estimate percentiles such as the 95^{th} percentile of the lognormal distribution.*

The point pattern in Output 17.2.2 has a slope of approximately 0.39 and an intercept of 5. The following statements reproduce this plot, adding a lognormal reference line with this slope and intercept. The result is shown in Output 17.2.4.

```
symbol v=plus;
title 'Lognormal Q-Q Plot for Diameters';
proc capability data=measures noprint;
   qqplot diameter / lognormal(sigma=0.5 theta=5 slope=0.39)
                     pctlaxis(grid)
                     vref  = 5.8 5.9 6.0;
run;
```

Output 17.2.4. Lognormal Q-Q Plot Identifying Percentiles



The PCTLAXIS option labels the major percentiles, and the GRID option draws percentile axis reference lines. The 95^{th} percentile is 5.9, since the intersection of

*You can also use a probability plot for this purpose. See Output 16.2.1 on page 459.

the distribution reference line and the 95^{th} reference line occurs at this value on the vertical axis.

Alternatively, you can compute this percentile from the estimated lognormal parameters. The $100\alpha^{\text{th}}$ percentile of the lognormal distribution is

$$P_\alpha = \exp(\sigma\Phi^{-1}(\alpha) + \zeta) + \theta$$

where $\Phi^{-1}(\cdot)$ is the inverse cumulative standard normal distribution. Consequently,

$$P_{0.95} \approx \exp\left(\frac{1}{2}\Phi^{-1}(0.95) + \log(0.39)\right) + 5 \approx \exp\left(\frac{1}{2} \times 1.645 - 0.94\right) + 5 \approx 5.89$$

Two-Parameter Lognormal Plots

See CAPQQ2 in the SAS/QC Sample Library

If a known threshold parameter is available, you can construct a two-parameter lognormal Q-Q plot by subtracting the threshold from the data and requesting a normal Q-Q plot. The following statements create this plot for DIAMETER, assuming a known threshold of five:

```
data measures;
   set measures;
   logdiam=log(diameter-5);
   label logdiam='log(Diameter-5)';
run;

symbol v=plus;
title 'Two-Parameter Lognormal Q-Q Plot for Diameters';
proc capability data=measures noprint;
   qqplot logdiam / normal(mu=est sigma=est)
                    square
                    vaxis=axis1;
   axis1 label=(a=90 r=0);
run;
```

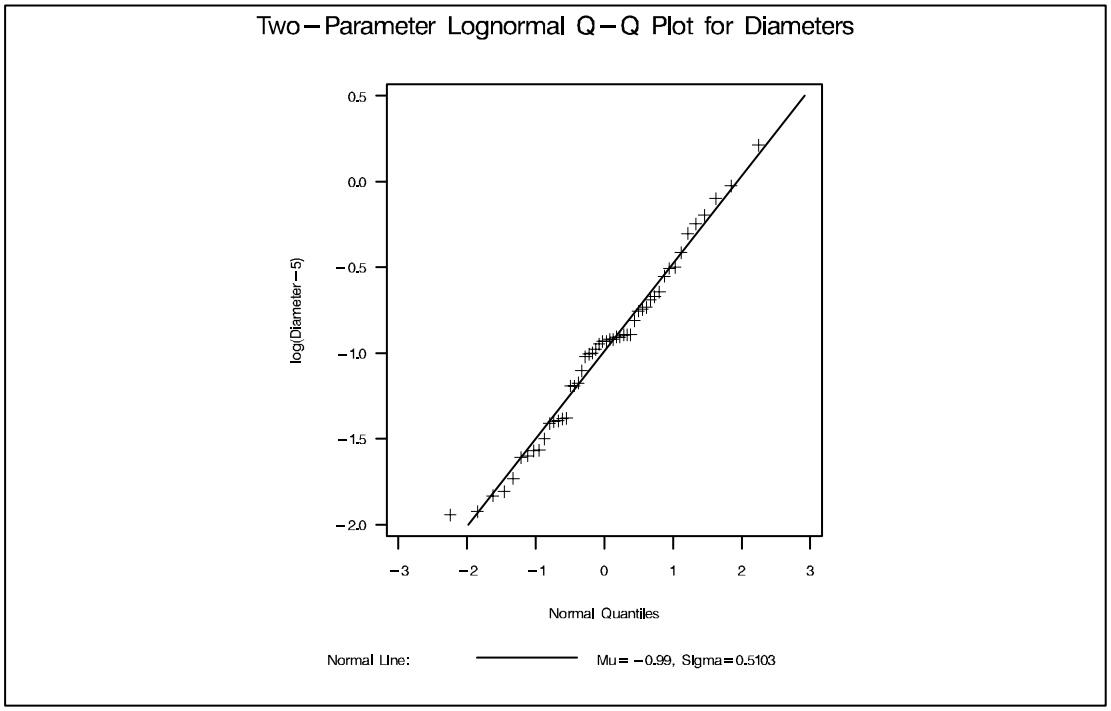
Because the point pattern in Output 17.2.5 is linear, you can estimate the lognormal parameters ζ and σ as the normal plot estimates of μ and σ, which are -0.99 and 0.51. These values correspond to the previous estimates of -0.92 for ζ and 0.5 for σ.

Example 17.3. Comparing Weibull Q-Q Plots

This example compares the use of three-parameter and two-parameter Weibull Q-Q plots for the failure times in months for 48 integrated circuits. The times are assumed to follow a Weibull distribution.

```
data failures;
   input time @@;
   label time='Time in Months';
   datalines;
29.42 32.14 30.58 27.50 26.08 29.06 25.10 31.34
29.14 33.96 30.64 27.32 29.86 26.28 29.68 33.76
29.32 30.82 27.26 27.92 30.92 24.64 32.90 35.46
```

Output 17.2.5. Two-Parameter Lognormal Q-Q Plot for Diameters




```
30.28 28.36 25.86 31.36 25.26 36.32 28.58 28.88
26.72 27.42 29.02 27.54 31.60 33.46 26.78 27.82
29.18 27.94 27.66 26.42 31.00 26.64 31.44 32.52
 ;
run;
```

Three-Parameter Weibull Plots

See CAPQQ3 in the SAS/QC Sample Library

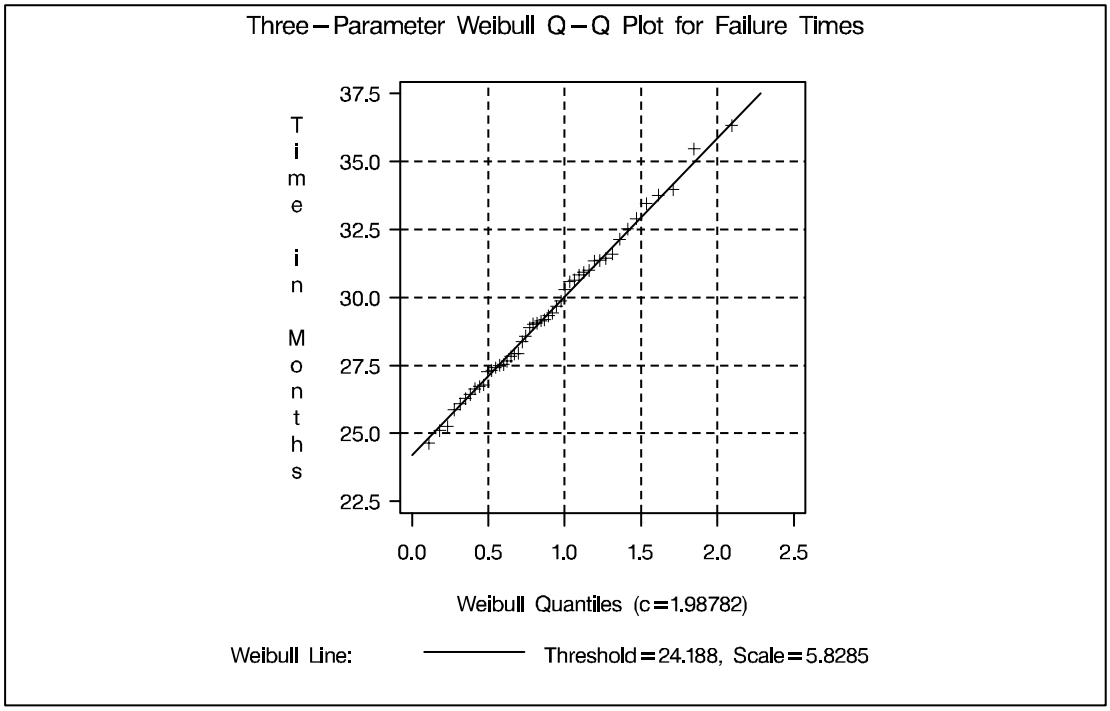
If no assumption is made about the parameters of this distribution, you can use the WEIBULL option to request a three-parameter Weibull plot. As in the previous example, you can visually estimate the shape parameter c by requesting plots for different values of c and choosing the value of c that linearizes the point pattern. Alternatively, you can request a maximum likelihood estimate for c, as illustrated in the following statements produce Weibull plots for $c = 1$, 2 and 3:

```
symbol v=plus;
title 'Three-Parameter Weibull Q-Q Plot for Failure Times';
proc capability data=failures noprint;
   qqplot time / weibull(c=est theta=est sigma=est)
                 square
                 href=0.5 1 1.5 2
                 vref=25 27.5 30 32.5 35;
run;
```

Note: When using the WEIBULL option, you must either specify a list of values for the Weibull shape parameter c with the C= option, or you must specify C=EST.

Output 17.3.1 displays the plot for the estimated value $c = 1.99$. The reference line corresponds to the estimated values for the threshold and scale parameters of ($\hat{\theta}_0$=24.19 and $\hat{\sigma}_0$=5.83, respectively.

Output 17.3.1. Three-Parameter Weibull Q-Q Plot for $c = 2$



Two-Parameter Weibull Plots

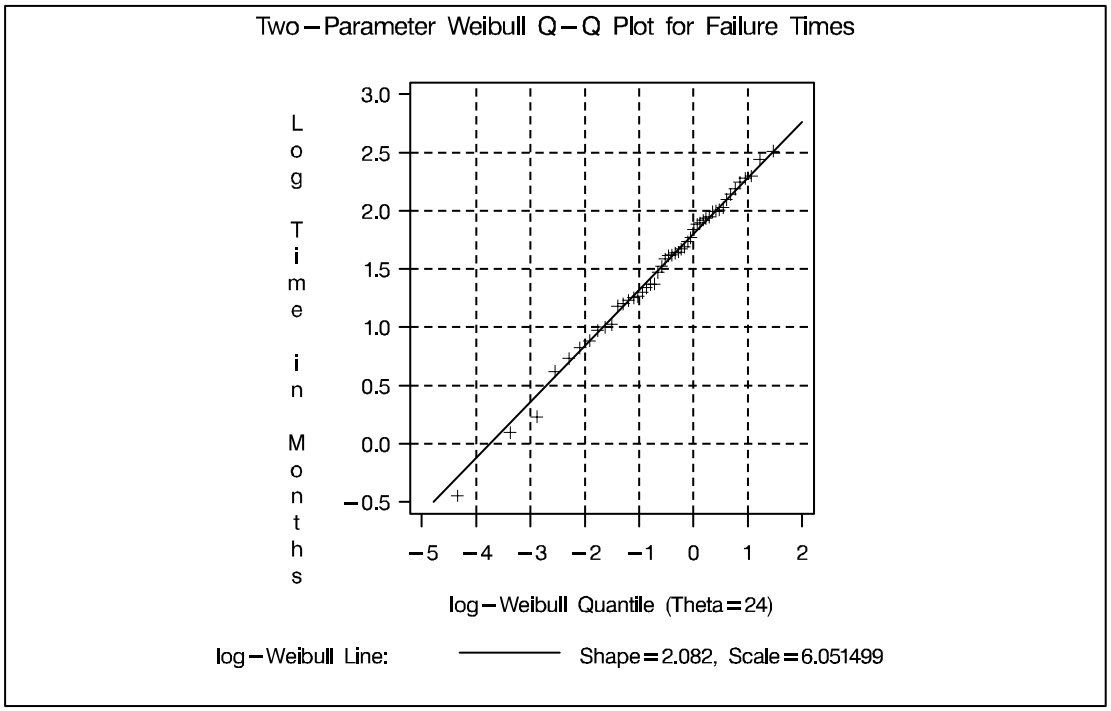
See CAPQQ3 in the SAS/QC Sample Library

Now, suppose it is known that the circuit lifetime is at least 24 months. The following statements use the threshold value $\theta_0 = 24$ to produce the two-parameter Weibull Q-Q plot shown in Output 17.3.2:

```
symbol v=plus;
title 'Two-Parameter Weibull Q-Q Plot for Failure Times';
proc capability data=failures noprint;
   qqplot time / weibull2(theta=24 c=est sigma=est) square
                 href= -4 to 1
                 vref= 0 to 2.5 by 0.5;
run;
```

The reference line is based on maximum likelihood estimates $\hat{c}$=2.08 and $\hat{\sigma}$=6.05. These estimates agree with those of the previous example.

Output 17.3.2. Two-Parameter Weibull Q-Q Plot for $\theta_0 = 24$



Example 17.4. Estimating Cpk from a Normal Q-Q Plot

See CAPQQ1 in the SAS/QC Sample Library

This example illustrates how you can use a normal Q-Q plot to estimate the capability index C_{pk}. The data used here are the distance measurements provided in the "Creating a Normal Quantile-Quantile Plot" section on page 464.

The linearity of the point pattern in Figure 17.3 on page 466 indicates that the measurements are normally distributed (recall that normality should be checked when process capability indices are reported). Furthermore, Figure 17.3 shows that the upper specification limit is about 1.7 standard deviation units above the mean, and the lower specification limit is about 1.8 standard deviation units below the mean. Since *CPU* is defined as

$$CPU = \frac{USL - \mu}{3\sigma}$$

and *CPL* is defined as

$$CPL = \frac{\mu - LSL}{3\sigma}$$

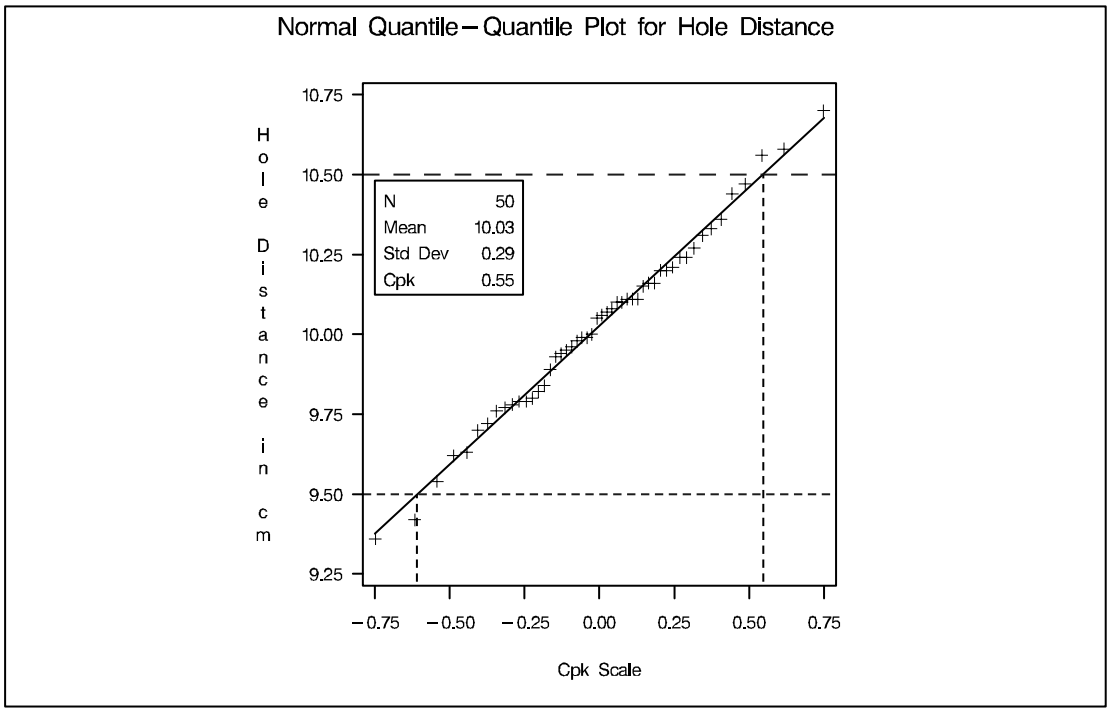
it follows that an estimate of *CPU* is $1.7/3 = 0.57$, and an estimate of *CPL* is $1.8/3 = 0.6$. Thus, except for a factor of three, you can estimate *CPU* and *CPL* from the points of intersection between the specification lines and the point pattern.

The following statements facilitate this type of estimation by creating a Q-Q plot, displayed in Output 17.4.1, in which the horizontal axis is rescaled by a factor of three:

```
symbol v=plus;
title "Normal Quantile-Quantile Plot for Hole Distance";
proc capability data=sheets noprint;
   spec  lsl=9.5   llsl=2   clsl=blue
         usl=10.5  lusl=20  cusl=blue;
   qqplot distance / normal(mu=est sigma=est cpkscale cpkref)
                nolegend
                square;
   inset n mean (5.2) std="Std Dev" (4.2) cpk (4.2) /
         pos=(-0.75,10.48) data refpoint=tl;
run;
```

The CPKSCALE option rescales the horizontal axis, and the CPKREF option adds reference lines indicating the intersections of the distribution reference line and the specification limits.

Output 17.4.1. Normal Q-Q Plot With C_{pk} Scaling



Using this display, you can estimate *CPU* and *CPL* directly from the horizontal axis as 0.55 and 0.60, respectively (the negative sign for -0.60 is ignored). The minimum of these values (0.55) is an estimate of C_{pk}. Note that this estimate agrees with the numerically obtained estimate for C_{pk} that is displayed on the plot with the INSET statement.

See Rodriguez (1992) for further discussion concerning the use of Q-Q plots in process capability analysis.

References

Anderson, T. W. and Darling, D. A. (1954), "A Test of Goodness-of-Fit," *Journal of the American Statistical Association*, 49, 765–769.

ASQC/AIAG Task Force (1991), *Fundamental Process Control, Reference Manual*, published by the Automotive Division of the American Society for Quality Control Supplier Quality Requirements Task Force, in collaboration with the Automotive Industry Action Group.

Bai, D. S. and Choi, I. S. (1997), "Process Capability Indices for Skewed Populations," *Manuscript*, Korean Advanced Institute of Science and Technology, Taejon, Korea.

Bissell, A. F. (1990), "How Reliable Is Your Capability Index?" *Applied Statistics*, 30, 331–340.

Blom, G. (1958), *Statistical Estimates and Transformed Beta Variables*, New York: John Wiley & Sons, Inc.

Bothe, D. R. (1997), *Measuring Process Capability*, New York: McGraw-Hill.

Bowman, K. O. and Shenton, L. R. (1983), "Johnson's System of Distributions," in *Encyclopedia of Statistical Sciences, Volume 4*, edited by S. Kotz, N. L. Johnson, and C. B. Read. New York: John Wiley & Sons, Inc., 303–314.

Boyles R. A. (1991), "The Taguchi Capability Index," *Journal of Quality Technology*, 23, 107–126.

Boyles, R. A. (1992), "C_{pm} for Asymmetrical Tolerances," *Technical Report*, Portland, OR: Precision Castparts Corp.

Boyles, R. A. (1994), "Process Capability with Asymmetric Tolerances," *Communication and Statistics, Part B–Simulation and Computation*, 23(3), 615–643.

Chambers, J. M., Cleveland, W. S., Kleiner, B., and Tukey, P. A. (1983), *Graphical Methods for Data Analysis*, Belmont, CA: Wadsworth International Group.

Chan, L. K., Cheng, S. W., and Spiring, F. A. (1990), "A New Measure of Process Capability: Cpm," *Journal of Quality Technology*, 30, 162–175.

Chan, L. K., Xiong, Z., and Zhang, D. (1990), "On the Asymptotic Distributions of Some Process Capability Indices," *Communications in Statistics–Theory and Methods*, 19, 11dashtwo18.

Chen, H. F. and Kotz, S. (1996), "An Asymptotic Distribution of Wright's Process Capability Index Sensitive to Skewness," *Journal of Statistical Computation and Simulation*, 55, 147–158.

Chen, K. S. (1998), "Incapability Index with Asymmetric Tolerances," *Statistica Sinica*, 8, 253–262.

Chou Y., Owen D. B., Borrego S. A. (1990), "Lower Confidence Limits on Process Capability Indices," *Journal of Quality Technology*, 22, 223–229. Corrigenda, 24, 251.

Cohen, A. C. (1951), "Estimating Parameters of Logarithmic-Normal Distributions by Maximum Likelihood," *Journal of the American Statistical Association*, 46, 206–212.

Conover, W. J. (1980), *Practical Nonparametric Statistics, 2nd Edition*, New York: John Wiley & Sons, Inc.

Croux, C. and Rousseeuw, P. J. (1992), "Time-Efficient Algorithms for Two Highly Robust Estimators of Scale," *Computational Statistics*, 1, 411–428.

D'Agostino, R. B. and Stephens, M. A., eds. (1986), *Goodness-of-Fit Techniques*, New York: Marcel Dekker, Inc.

David, H. A. (1981), *Order Statistics, Second Edition*, New York: John Wiley & Sons, Inc.

Dixon, W. J. and Tukey, J. W. (1968), "Approximate Behavior of the Distribution of Winsorized t (Trimming/Winsorization 2)," *Technometrics*, 10, 83–98.

Elderton, W. P. and Johnson, N. L. (1969), *Systems of Frequency Curves*, Cambridge: University Press.

Ekvall, D. N. and Juran, J. M. (1974), "Manufacturing Planning," *Quality Control Handbook, Third Edition*, New York: McGraw-Hill.

Elandt, R. C. (1961), "The Folded Normal Distribution: Two Methods of Estimating Parameters from Moments," *Technometrics*, 3, 551–562.

Fisher, R. A. (1973), *Statistical Methods for Research Workers, 14th Edition*, New York: Hafner Publishing Company.

Fowlkes, E. B. (1987), *A Folio of Distributions: A Collection of Theoretical Quantile-Quantile Plots*, New York: Marcel Dekker, Inc.

Gnanadesikan, R. (1997), *Statistical Data Analysis of Multivariate Observations*, New York: John Wiley & Sons, Inc.

Guirguis, G. H. and Rodriguez, R. N. (1992), "Computation of Owen's Q Function Applied to Process Capability Analysis," *Journal of Quality Technology*, 24, 236–246.

Gupta, A. K. and Kotz, S. (1997), "A New Process Capability Index," *Metrika*, 45, 213–224.

Hahn, G. J. (1969), "Factors for Calculating Two-Sided Prediction Intervals for Samples from a Normal Distribution," *Journal of the American Statistical Association*, 64, 878–898.

Hahn, G. J. (1970a), "Additional Factors for Calculating Prediction Intervals for Samples from a Normal Distribution," *Journal of the American Statistical Association*, 65, 1668–1676.

Hahn, G. J. (1970b), "Statistical Intervals for a Normal Population, Part I. Tables, Examples and Applications," *Journal of Quality Technology*, 2, 115–125.

Hahn, G. J. (1970c), "Statistical Intervals for a Normal Population, Part II. Formulas, Assumptions, Some Derivations," *Journal of Quality Technology*, 2, 115–125.

Hahn, G. J. and Meeker, W. Q. (1991), *Statistical Intervals: A Guide for Practitioners*, New York: John Wiley & Sons, Inc.

Hampel, F. R. (1974), "The Influence Curve and Its Role in Robust Estimation," *Journal of the American Statistical Association*, 69, 383–393.

Iman, R. L. (1974), "Use of a t-statistic as an Approximation to the Exact Distribution of the Wilcoxon Signed Rank Statistic," *Communications in Statistics*, 3, 795–806.

Johnson, N. L., Kotz, S., and Balakrishnan, N. (1994), *Continuous Univariate Distributions–1, Second Edition*, New York: John Wiley & Sons, Inc.

Johnson, N. L., Kotz, S., and Balakrishnan, N. (1995), *Continuous Univariate Distributions–2, Second Edition*, New York: John Wiley & Sons, Inc.

Johnson, N. L., Kotz, S., and Pearn, W. L. (1994), "Flexible Process Capability Indices," *Pakistan Journal of Statistics*, 10(1)A, 23–31.

Kane, V. E. (1986), "Process Capability Indices," *Journal of Quality Technology*, 1, 41–52.

Kotz, S. and Johnson, N. L. (1993), *Process Capability Indices*, London: Chapman & Hall.

Kotz, S. and Lovelace, C. R. (1998), *Process Capability Indices in Theory and Practice*, London: Arnold.

Kushler, R. H. and Hurley, P. (1992), "Confidence Bounds for Capability Indices," *Journal of Quality Technology*, 24, 188–195.

Lehmann, E. L. (1975), *Nonparametrics: Statistical Methods Based on Ranks*, San Francisco: Holden-Day, Inc.

Luceño, A. (1996), "A Process Capability Index with Reliable Confidence Intervals," *Communications in Statistics – Simulation*, 25(1), 235–245.

Mage, D. T. (1980), "An Explicit Solution for S_B Parameters Using Four Percentile Points," *Technometrics*, 22, 247–251.

Marcucci, M. O. and Beazley, C. F. (1988), "Capability Indices: Process Performance Measures," *Transactions of ASQC Congress*, 516–523.

Montgomery, D. (1996), *Introduction to Statistical Quality Control, Third Edition*, New York: John Wiley & Sons, Inc.

Mood, A. M., Graybill, F. A., and Boes, D. C. (1974), *Introduction to the Theory of Statistics, Third Edition*, New York: McGraw-Hill.

Odeh, R. E. and Owen, D. B. (1980), *Tables for Normal Tolerance Limits, Sampling Plans, and Screening*, New York: Marcel Dekker, Inc.

Owen, D. B. and Hua, T. A. (1977), "Tables of Confidence Limits on the Tail Area of the Normal Distribution," *Communication and Statistics, Part B–Simulation and Computation*, 6, 285–311.

Parzen, E. (1979), "Nonparametric Statistical Data Modeling," *Journal of the American Statistical Association*, 71, 105–121.

Pearn, W. L., Kotz, S., and Johnson, N. L. (1992), "Distributional and Inferential Properties of Process Capability Indices," *Journal of Quality Technology*, 24, 216–231.

Rodriguez, R. N. (1992), "Recent Developments in Process Capability Analysis," *Journal of Quality Technology*, 24, 176–187.

Rousseeuw, P. J. and Croux, C. (1993), "Alternatives to the Median Absolute Deviation," *Journal of the American Statistical Association*, 88, 1273–1283.

Royston, J. P. (1982), "An Extension of Shapiro and Wilk's W Test for Normality to Large Samples," *Applied Statistics*, 131, 115–124.

Royston, J. P. (1992), "Approximating the Shapiro-Wilk's W Test for Nonnormality," *Statistics and Computing,* 2, 117-119.

SAS Institute Inc. (1999), *SAS/GRAPH Software: Reference, Version 8*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1999), *SAS Language Reference: Dictionary, Version 8*, Cary, NC: SAS Institute Inc.

Shapiro, S. S. and Wilk, M. B. (1965), "An Analysis of Variance Test for Normality (Complete Samples)," *Biometrika*, 52, 591–611.

Silverman, B. W. (1986), *Density Estimation for Statistics and Data Analysis*, New York: Chapman and Hall.

Slifker, J. F. and Shapiro, S. S. (1980), "The Johnson System: Selection and Parameter Estimation," *Technometrics*, 22, 239–246.

Sprent, P. (1989), *Applied Nonparametric Statistical Methods*, New York: Chapman and Hall.

Stephens, M. A. (1974), "EDF Statistics for Goodness of Fit and Some Comparisons," *Journal of the American Statistical Association*, 69, 730–737.

Terrell, G. R. and Scott, D. W. (1985), "Oversmoothed Nonparametric Density Estimates," *Journal of the American Statistical Association*, 80, 209–214.

Tukey, J. W. (1977), *Exploratory Data Analysis*, Reading, MA: Addison-Wesley Publishing Co., Inc.

Tukey, J. W. and McLaughlin, D. H. (1963), "Less Vulnerable Confidence and Significance Procedures for Location Based on a Single Sample: Trimming/Winsorization 1," *Sankhya A*, 25, 331–352.

Vännmann, K. (1995), "A Unified Approach to Capability Indices," *Statistica Sinica*, 5(2), 805–820.

Vännmann, K. (1997), "A General Class of Capability Indices in the Case of Asymmetric Tolerances," *Communications in Statistics – Theory and Methods*, 26(8), 2049–2072.

Velleman, P. F. and Hoaglin, D. C. (1981), *Applications, Basics, and Computing of Exploratory Data Analysis*, Boston, MA: Duxbury Press.

Wadsworth, H. M., Stephens, K. S., and Godfrey, A. B. (1986), *Modern Methods for Quality Control and Improvement*, New York: John Wiley & Sons, Inc.

Wainer, H. (1974), "The Suspended Rootogram and Other Visual Displays: An Empirical Validation," *The American Statistician*, 28, 143–145.

Wilk, M. B. and Gnanadesikan, R. (1968), "Probability Plotting Methods for the Analysis of Data," *Biometrika*, 49, 525–545.

Wright, P. A. (1995), "A Process Capability Index Sensitive to Skewness," *Journal of Statistical Computation and Simulation,* 52, 195–203.

Zhang, N. F., Stenback, G. A., Wardrop, D. M. (1990), "Interval Estimation of Process Capability Index Cpk," *Comm. in Stat. Theory and Methods*, 19, 4455–4470.

Part 3
The CUSUM Procedure

Contents

Introduction 509

Chapter 18. PROC CUSUM Statement 511

Chapter 19. XCHART Statement 519

Chapter 20. INSET Statement 577

References 583

Introduction

The CUSUM procedure creates cumulative sum control charts, also known as *cusum charts*, which display cumulative sums of the deviations of measurements or subgroup means from a target value. Cusum charts are used to decide whether a process is in statistical control by detecting a shift in the process mean.

You can use the CUSUM procedure to

- apply a *one-sided cusum scheme*, also referred to as a *decision interval scheme*, which detects a shift in one direction from the target mean. You can specify the scheme with the decision interval h and the reference value k.
- apply a *two-sided cusum scheme* with a V-mask, which detects a shift in either direction from the target mean. You can specify the scheme with geometric parameters (h and k) for the V-mask or with error probabilities (α and β).
- implement cusum schemes graphically or computationally
- specify the shift to be detected as a multiple of standard error or in data units
- estimate the process standard deviation σ using a variety of methods
- compute average run lengths (ARLs)
- read raw data (actual measurements) or summarized data (subgroup means and standard deviations)
- analyze multiple process variables. If used with a BY statement, PROC CUSUM produces charts separately for groups of observations.
- save cusums and cusum scheme parameters in output data sets
- tabulate the information displayed on the chart
- read cusum scheme parameters from an input data set
- read numeric- or character-valued subgroup variables
- display subgroups with date and time formats
- enhance cusum charts with special legends and symbol markers that indicate the levels of stratification variables
- superimpose plotted points with stars (polygons) whose vertices indicate the values of multivariate data related to the process
- display a trend chart below the cusum chart that plots a systematic or fitted trend in the data
- display charts on line printers or on graphics devices. Charts produced on line printers can use special formatting characters that improve the appearance of the chart. Charts produced on graphics devices can be annotated, saved, and replayed.

Learning about the CUSUM Procedure

If you are using the CUSUM procedure for the first time, begin by reading Chapter 18, "PROC CUSUM Statement," to learn about input data sets. Then turn to "Getting Started" on page 521 in Chapter 19, "XCHART Statement." This chapter also provides syntax information and advanced examples.

If you are not familiar with cusum charts, read "Formulas for Cumulative Sums," "Defining the Decision Interval for a One-Sided Cusum Scheme," and "Defining the V-Mask for a Two-Sided Cusum Scheme" in the "Details" section on page 551. References lists articles and textbooks that provide more detailed information on cusum charts. The expository articles by Lucas (1976) and Goel (1982) and the textbooks by Montgomery (1996) and Ryan (1989) are recommended introductory reading.

Chapter 18
PROC CUSUM Statement

Chapter Contents

OVERVIEW 513

SYNTAX 514
Input and Output Data Sets 517

Chapter 18
PROC CUSUM Statement

Overview

The PROC CUSUM statement starts the CUSUM procedure and it identifies input data sets.

After the PROC CUSUM statement, you provide an XCHART statement that specifies the cusum chart you want to create and the variables in the input data set that you want to analyze. For example, the following statements request a one-sided (decision interval) cusum chart:

```
proc cusum data=values;
   xchart weight*lot / scheme = onesided
                       mu0    = 8.100
                       sigma0 = 0.050
                       delta  = 1
                       h      = 2.2
                       k      = 0.5;
run;
```

In this example, the DATA= option specifies an input data set (VALUES) that contains the *process* measurement variable WEIGHT and the *subgroup-variable* LOT. *

You can use options in the PROC CUSUM statement to

- specify input data sets containing variables to be analyzed, parameters for cusum schemes, or annotation information
- specify a graphics catalog for saving graphical output
- specify that charts are to be produced on graphics devices or line printers
- define characters used for features on charts produced on line printers

In addition to the XCHART statement, you can provide BY statements, ID statements, TITLE statements, and FOOTNOTE statements. If you are using a graphics device, you can also provide graphics enhancement statements, such as SYMBOLn statements, which are described in *SAS/GRAPH Software: Reference*.

Note: If you are using the CUSUM procedure for the first time, you should read both this chapter and "Getting Started" on page 521 in Chapter 19, "XCHART Statement."

*In Release 6.12 and previous releases of SAS/QC software, the keyword GRAPHICS was required in the PROC CUSUM statement to specify that the chart be created with a graphics device. In Version 7, you can specify the LINEPRINTER option to request line printer plots.

Syntax

The syntax for the PROC CUSUM statement is as follows:

PROC CUSUM < *options* >**;**

The PROC CUSUM statement starts the CUSUM procedure, and it optionally identifies various data sets and requests graphics output. You can specify the following *options* in the PROC CUSUM statement. The marginal notes *Graphics* and *Line Printer* identify options that apply to graphics devices and line printers, respectively.

ANNOTATE=*SAS-data-set*
ANNO=*SAS-data-set*

Graphics

specifies an input data set that contains appropriate annotate variables, as described in *SAS/GRAPH Software: Reference*. The ANNOTATE= option allows you to add features to the cusum chart (for example, labels that explain out-of-control points). The ANNOTATE= data set is used only when the chart is created using a graphics device; it is ignored when the LINEPRINTER option is specified. The data set specified with the ANNOTATE= option in the PROC CUSUM statement is a "global" annotate data set in the sense that the information in this data set is displayed on every chart produced in the current run of the CUSUM procedure.

ANNOTATE2=*SAS-data-set*
ANNO2=*SAS-data-set*

Graphics

specifies an input data set that contains appropriate annotate variables that add features to the trend chart (secondary chart) produced with the TRENDVAR= option in the XCHART statement.

DATA=*SAS-data-set*

names an input data set that contains raw data (measurements) as observations. If the values of the *subgroup-variable* are numeric, you need to sort the data set so that these values are in increasing order (within BY groups). The DATA= data set can contain more than one observation for each value of the *subgroup-variable*.

You cannot use a DATA= data set with a HISTORY= data set. If you do not specify a DATA= or HISTORY= data set, PROC CUSUM uses the most recently created data set as a DATA= data set. For more information, see "DATA= Data Set" on page 566.

FORMCHAR(*index***)='***string***'**

Line Printer

defines characters used for features on charts produced on a line printer, where

index

is a list of numbers ranging from 1 to 17. The list identifies which features are controlled with the *string* characters. By default, *index* is omitted, and the FORMCHAR= option gives a *string* for all 17 features.

string

gives characters for features in *index*. Any character or hexadecimal string can be used.

The features associated with values of *index* are as follows:

Value of *index*	Description of Character	Chart Feature
1	vertical bar	frame
2	horizontal bar	frame, central line
3	box character (upper left)	frame
4	box character (upper middle)	serifs, tick (horizontal axis)
5	box character (upper right)	frame
6	box character (middle left)	not used
7	box character (middle middle)	serifs
8	box character (middle right)	tick (vertical axis)
9	box character (lower left)	frame
10	box character (lower middle)	serifs
11	box character (lower right)	frame
12	vertical bar	control limits
13	horizontal bar	control limits
14	box character (upper right)	control limits
15	box character (lower left)	control limits
16	box character (lower right)	control limits
17	box character (upper left)	control limits

Not all printers can produce the characters in the preceding list. By default, the form character list specified by the SAS system option FORMCHAR= is used; otherwise, the default is FORMCHAR='|—-|+|—|=====’. If you print to a PC screen or if your device supports the ASCII symbol set (1 or 2), the following is recommended:

```
formchar='B3,C4,DA,C2,BF,C3,C5,B4,C0,C1,D9,BA,CD,BB,C8,BC,D9'X
```

Note that you can use the FORMCHAR= option to temporarily override the values of the SAS system FORMCHAR= option. The values of the SAS system FORMCHAR= option are not altered by the FORMCHAR= option in the PROC CUSUM statement.

GOUT=*graphics-catalog*

specifies the graphics catalog for graphics output from PROC CUSUM. This is useful if you want to save the output. The GOUT= option is used only when the chart is created using a graphics device; it is ignored when the LINEPRINTER option is specified. *Graphics*

HISTORY=*SAS-data-set*
HIST=*SAS-data-set*

names an input data set that contains subgroup summary statistics (means, standard deviations, and sample sizes). Typically, this data set is created as an OUTHISTORY= data set in a previous run of PROC CUSUM or PROC SHEWHART, but it can also be created with a SAS summarization procedure such as PROC MEANS.

If the values of the *subgroup-variable* are numeric, you need to sort the data set so that these values are in increasing order (within BY groups). A HISTORY= data set can contain only one observation for each value for the *subgroup-variable*.

You cannot use a HISTORY= data set together with a DATA= data set. If you do not specify a HISTORY= or DATA= data set, PROC CUSUM uses the most recently created data set as a DATA= data set. For more information on HISTORY= data sets, see "HISTORY= Data Set" on page 568.

LIMITS=*SAS-data-set*

names an input data set that contains a set of decision interval or V-mask parameters. Each observation in a LIMITS= data set contains the parameters for a *process*.

If you are using Release 6.09 or an earlier release of SAS/QC software, you must specify the options READLIMITS or READINDEX= in the XCHART statement to read the parameters from the LIMITS= data set. In Release 6.10 and later releases, these options are not needed.

For details about the variables needed in a LIMITS= data set, see "LIMITS= Data Set" on page 567. If you do not provide a LIMITS= data set, you must specify the parameters with options in the XCHART statement.

LINEPRINTER

requests that line printer charts be produced. By default, the procedure creates charts for a graphics device.

Input and Output Data Sets

Figure 18.1 summarizes the data sets used with the CUSUM procedure.

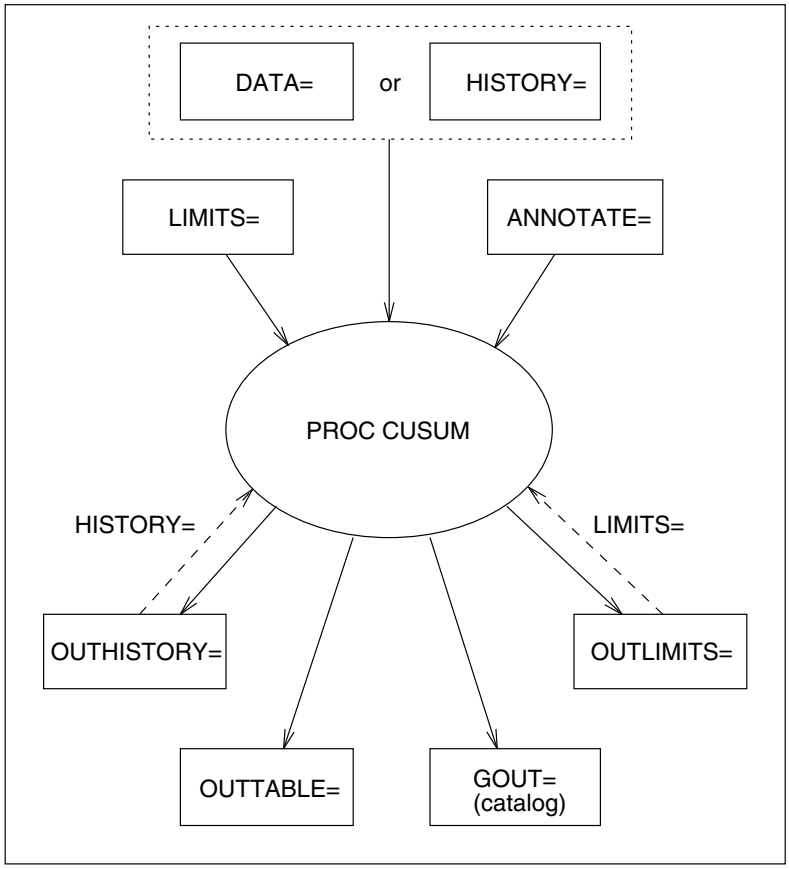


Figure 18.1. Input and Output Data Sets in the CUSUM Procedure

Chapter 19
XCHART Statement

Chapter Contents

OVERVIEW 521

GETTING STARTED 521
- Creating a V-Mask Cusum Chart from Raw Data 522
- Creating a V-Mask Cusum Chart from Subgroup Summary Data 525
- Saving Summary Statistics 527
- Creating a One-Sided Cusum Chart with a Decision Interval 528
- Saving Cusum Scheme Parameters 531
- Reading Cusum Scheme Parameters 533

SYNTAX 535
- Summary of Options 537
- Dictionary of Special Options 544

DETAILS 551
- Basic Notation for Cusum Charts 551
- Formulas for Cumulative Sums 552
- Defining the Decision Interval for a One-Sided Cusum Scheme 554
- Defining the V-Mask for a Two-Sided Cusum Scheme 555
- Designing a Cusum Scheme 557
- Cusum Charts Compared with Shewhart Charts 560
- Methods for Estimating the Standard Deviation 561
- Output Data Sets 564
- ODS Tables 566
- Input Data Sets 566
- Missing Values 569

EXAMPLES 569
- Example 19.1. Cusum and Standard Deviation Charts 569
- Example 19.2. Upper and Lower One-Sided Cusum Charts 572
- Example 19.3. Combined Shewhart–Cusum Scheme 574

Chapter 19
XCHART Statement

Overview

The XCHART statement creates cumulative sum control charts from subgroup means or individual measurements. You can create these charts for one-sided cusum (decision interval) schemes or for two-sided (V-mask) schemes. A one-sided scheme is designed to detect either a positive or a negative shift from the target mean, and a two-sided scheme is designed to detect positive and negative shifts from the target mean.

You can use options in the XCHART statement to

- specify parameters for a decision interval or V-mask
- specify the shift δ to be detected
- specify the target mean μ_0
- specify a known (standard) value σ_0 for the process standard deviation or estimate the standard deviation from the data using various methods
- tabulate the information displayed on the chart
- save the information displayed on the chart in an output data set
- read parameters for the cusum scheme from a data set
- display a secondary chart that plots a time trend that has been removed from the data
- add block legends and special symbol markers to reveal stratification in process data
- superimpose stars at each point to represent related multivariate factors
- display vertical and horizontal reference lines
- modify the axis values and labels
- modify the chart layout and appearance

Getting Started

This section introduces the XCHART statement with simple examples that illustrate the most commonly used options. Complete syntax for the XCHART statement is presented in the "Syntax" section on page 535, and advanced examples are given in the "Examples" section on page 569.

Creating a V-Mask Cusum Chart from Raw Data

See CUSTWOS1 in the SAS/QC Sample Library

A machine fills eight-ounce cans of two-cycle engine oil additive. The filling process is believed to be in statistical control, and the process is set so that the average weight of a filled can is $\mu_0 = 8.100$ ounces. Previous analysis shows that the standard deviation of fill weights is $\sigma_0 = 0.050$ ounces. A two-sided cusum chart is used to detect shifts of at least one standard deviation in either the positive or negative direction from the target mean of 8.100 ounces.

Subgroup samples of four cans are selected every hour for twelve hours. The cans are weighed, and their weights are saved in a SAS data set named OIL.

```
data oil;
   label hour = 'Hour';
   input hour @;
   do i=1 to 4;
      input weight @;
      output;
      end;
   drop i;
   datalines;
 1  8.024  8.135  8.151  8.065
 2  7.971  8.165  8.077  8.157
 3  8.125  8.031  8.198  8.050
 4  8.123  8.107  8.154  8.095
 5  8.068  8.093  8.116  8.128
 6  8.177  8.011  8.102  8.030
 7  8.129  8.060  8.125  8.144
 8  8.072  8.010  8.097  8.153
 9  8.066  8.067  8.055  8.059
10  8.089  8.064  8.170  8.086
11  8.058  8.098  8.114  8.156
12  8.147  8.116  8.116  8.018
;
run;

proc print data=oil noobs;
run;
```

The data set OIL is partially listed in Figure 19.1.

Each observation contains one value of WEIGHT along with its associated value of HOUR, and the values of HOUR are in increasing order. The CUSUM procedure assumes that DATA= input data sets are sorted in this "strung-out" form.

The following statements request a two-sided cusum chart with a V-mask for the average weights:

hour	weight
1	8.024
1	8.135
1	8.151
1	8.065
2	7.971
2	8.165
2	8.077
2	8.157
3	8.125
3	8.031
3	8.198
3	8.050
4	8.123
.	.
.	.
.	.
12	8.018

Figure 19.1. Partial Listing of the Data Set OIL

```
title 'Cusum Chart for Average Weights of Cans';
proc cusum data=oil;
   xchart weight*hour /
      mu0    = 8.100           /* Target mean for process  */
      sigma0 = 0.050           /* Known standard deviation */
      delta  = 1               /* Shift to be detected     */
      alpha  = 0.10            /* Type I error probability */
      vaxis  = -5 to 3 ;
   label weight = 'Cumulative Sum';
run;
```

The CUSUM procedure is invoked with the PROC CUSUM statement. The DATA= option in the PROC CUSUM statement specifies that the SAS data set OIL is to be read. The variables to be analyzed are specified in the XCHART statement. The process measurement variable (WEIGHT) is specified before the asterisk (this variable is referred to more generally as a *process*). The time variable (HOUR) is specified after the asterisk (this variable is referred to more generally as a *subgroup-variable* because it determines how the measurements are classified into rational subgroups).

The option ALPHA=0.10 specifies the probability of a Type 1 error for the cusum scheme (the probability of detecting a shift when none occurs).

The cusum chart is shown in Figure 19.2.

The cusum S_1 plotted at HOUR=1 is simply the standardized deviation of the first subgroup mean from the target mean.

$$S_1 = \frac{8.09375 - 8.100}{0.050/\sqrt{4}} = -0.250$$

The cusum S_2 plotted at HOUR=2 is S_1 plus the standardized deviation of the second

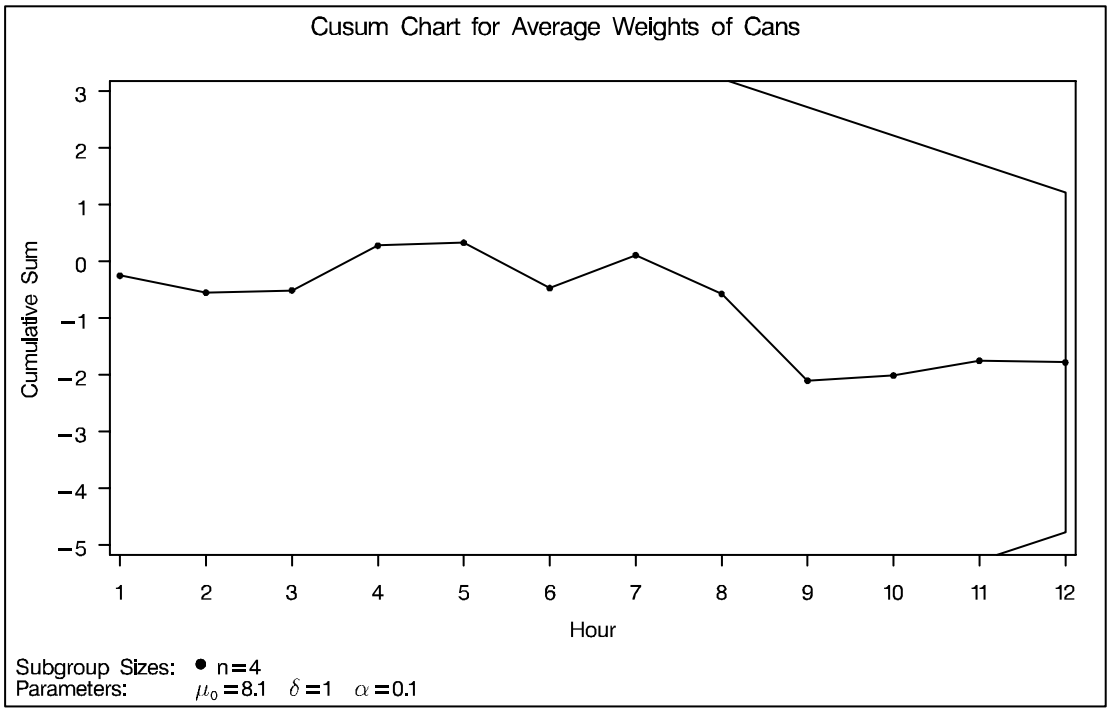


Figure 19.2. Two-Sided Cusum Chart with V-Mask

subgroup mean from the target mean.

$$S_2 = S_1 + \frac{8.0925 - 8.100}{0.050/\sqrt{4}} = -0.550$$

In general, the cusum plotted at HOUR=t is S_{t-1} plus the standardized deviation of the t^{th} subgroup mean from the target mean.

$$S_t = S_{t-1} + \frac{\bar{X}_t - \mu_0}{\sigma_0/\sqrt{n}}$$

For further details, see "Two-Sided Cusum Schemes" on page 553.

You can interpret the chart by comparing the points with the V-mask whose right edge is centered at the most recent point (HOUR=12). Since none of the points cross the arms of the V-mask, there is no evidence that a shift has occurred, and the fluctuations in the cusums can be attributed to chance variation. In general, crossing the lower arm is evidence of an increase in the process mean, whereas crossing the upper arm is evidence of a decrease in the mean.

Creating a V-Mask Cusum Chart from Subgroup Summary Data

See CUSTWOS1 in the SAS/QC Sample Library

The previous example illustrates how you can create a cusum chart using raw process measurements read from a DATA= data set. In many applications, however, the data are provided in *summarized form* as subgroup means. This example illustrates the use of the XCHART statement when the input data set is a HISTORY= data set.

The following data set provides the subgroup means, standard deviations, and sample sizes corresponding to the variable WEIGHT in the data set OIL (see page 522):

```
data oilstat;
   label hour = 'Hour';
   input hour weightx weights weightn;
   datalines;
 1  8.0938  0.0596  4
 2  8.0925  0.0902  4
 3  8.1010  0.0763  4
 4  8.1198  0.0256  4
 5  8.1013  0.0265  4
 6  8.0800  0.0756  4
 7  8.1145  0.0372  4
 8  8.0830  0.0593  4
 9  8.0618  0.0057  4
10  8.1023  0.0465  4
11  8.1065  0.0405  4
12  8.0993  0.0561  4


title;
proc print;
run;
```

The data set OILSTAT is listed in Figure 19.3.

Obs	hour	weightx	weights	weightn
1	1	8.0938	0.0596	4
2	2	8.0925	0.0902	4
3	3	8.1010	0.0763	4
4	4	8.1198	0.0256	4
5	5	8.1013	0.0265	4
6	6	8.0800	0.0756	4
7	7	8.1145	0.0372	4
8	8	8.0830	0.0593	4
9	9	8.0618	0.0057	4
10	10	8.1023	0.0465	4
11	11	8.1065	0.0405	4
12	12	8.0993	0.0561	4

Figure 19.3. Listing of the Data Set OILSTAT

Since the data set contains a subgroup variable, a mean variable, a standard deviation variable, and a sample size variable, it can be read as a HISTORY= data set. Note that

the names WEIGHTX, WEIGHTS, and WEIGHTN satisfy the naming conventions for summary variables since they begin with a common prefix (WEIGHT) and end with the suffix letters *X*, *S*, and *N*.

The following statements create the cusum chart:

```
title 'Cusum Chart for Average Weights of Cans';

proc cusum history=oilstat;
   xchart weight*hour /
      mu0    = 8.100          /* target mean               */
      sigma0 = 0.050          /* known standard deviation */
      delta  = 1              /* shift to be detected     */
      alpha  = 0.10           /* Type 1 error probability */
      vaxis  = -5 to 3 ;
   label weightx = 'Cumulative Sum';
run;
```

Note that the *process* WEIGHT specified in the XCHART statement is the prefix of the summary variable names in PARTS. Also note that the vertical axis label is specified by associating a variable label with the subgroup mean variable (PARTGAPX). The horizontal axis label is specified by associating a variable label with the *subgroup-variable* (HOUR). The chart (not shown here) is identical to the one in Figure 19.1.

In general, a HISTORY= input data set used with the XRCHART statement must contain the following four variables:

- subgroup variable
- subgroup mean variable
- subgroup range variable
- subgroup sample size variable

Furthermore, the names of subgroup mean, standard deviation, and sample size variables must begin with the prefix *process* specified in the XRCHART statement and end with the special suffix characters *X*, *S*, and *N*, respectively.

Note that the interpretation of *process* depends on the input data set specified in the PROC CUSUM statement.

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names containing the summary statistics.

For more information, see "DATA= Data Set" on page 566 and "HISTORY= Data Set" on page 568.

Saving Summary Statistics

See CUSTWOS1 in the SAS/QC Sample Library

In this example, the CUSUM procedure is used to save summary statistics and cusums in an output data set. The summary statistics can subsequently be analyzed by the CUSUM procedure (as in the preceding example). The following statements read the raw measurements from the data set OIL (see page 522) and create a summary data set named OILHIST:

```
title 'Cusum Chart for Average Weights of Cans';
proc cusum data=oil;
   xchart weight*hour /
      nochart
      outhistory = oilhist
      mu0        = 8.100        /* Target mean for process  */
      sigma0     = 0.050        /* Known standard deviation */
      delta      = 1            /* Shift to be detected     */
      alpha      = 0.10         /* Type I error probability */
      vaxis      = -5 to 3 ;
   label weight = 'Cumulative Sum';
run;



proc print data=oilhist;
   format weightx weights weightc 6.4 ;
run;
```

The OUTHISTORY= option names the SAS data set containing the summary information, and the NOCHART option suppresses the display of the charts (since the purpose here is simply to create an output data set). Figure 19.4 lists the data set OILHIST.

Obs	hour	weight X	weight S	weight C	weight N
1	1	8.0938	0.0596	-.2500	4
2	2	8.0925	0.0902	-.5500	4
3	3	8.1010	0.0763	-.5100	4
4	4	8.1198	0.0256	0.2800	4
5	5	8.1013	0.0265	0.3300	4
6	6	8.0800	0.0756	-.4700	4
7	7	8.1145	0.0372	0.1100	4
8	8	8.0830	0.0593	-.5700	4
9	9	8.0618	0.0057	-2.100	4
10	10	8.1023	0.0465	-2.010	4
11	11	8.1065	0.0405	-1.750	4
12	12	8.0993	0.0561	-1.780	4

Figure 19.4. Listing of the Data Set OILHIST

There are five variables in the data set.

- HOUR contains the subgroup index
- WEIGHTX contains the subgroup means

- WEIGHTS contains the subgroup standard deviations
- WEIGHTC contains the cumulative sums
- WEIGHTN contains the subgroup sample sizes

Note that the variables in the OUTHISTORY= data set are named by adding the suffix characters *X*, *S*, *N*, and *C* to the *process* WEIGHT specified in the XCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 565.

Creating a One-Sided Cusum Chart with a Decision Interval

See CUSONES1 in the SAS/QC Sample Library

An alternative to the V-mask cusum chart is the one-sided cusum chart with a decision interval, which is sometimes referred to as the "computational form of the cusum chart." This example illustrates how you can create a one-sided cusum chart for individual measurements.

A can of oil is selected every hour for fifteen hours. The cans are weighed, and their weights are saved in a SAS data set named CANS:*

```
data cans;
   length comment $16;
   label hour = 'Hour';
   input hour weight comment $16. ;
   datalines;
 1  8.024
 2  7.971
 3  8.125
 4  8.123
 5  8.068
 6  8.177  Pump Adjusted
 7  8.229  Pump Adjusted
 8  8.072
 9  8.066
10  8.089
11  8.058
12  8.147
13  8.141
14  8.047
15  8.125
;
run;
```

Suppose the problem is to detect a *positive* shift in the process mean of one standard deviation ($\delta = 1$) from the target of 8.100 ounces. Furthermore, suppose that

- a known value $\sigma_0 = 0.050$ is available for the process standard deviation
- an in-control average run length (ARL) of approximately 100 is required
- an ARL of approximately five is appropriate for detecting the shift

*This data set is used by later examples in this chapter.

Table 19.18 on page 559 indicates that these ARLs can be achieved with the decision interval $h = 3$ and the reference value $k = 0.5$. The following statements use these parameters to create the chart and tabulate the cusum scheme:

```
title "One-Sided Cusum Analysis";
proc cusum data=cans;
   xchart weight*hour /
      mu0    = 8.100     /* target mean for process     */
      sigma0 = 0.050     /* known standard deviation    */
      delta  = 1         /* shift to be detected        */
      h      = 3         /* cusum parameter h           */
      k      = 0.5       /* cusum parameter k           */
      scheme = onesided  /* one-sided decision interval */
      tableall           /* table                       */
      ;
   label weight = 'Cusum of Weight';
run;
```

The chart is shown in Figure 19.5.

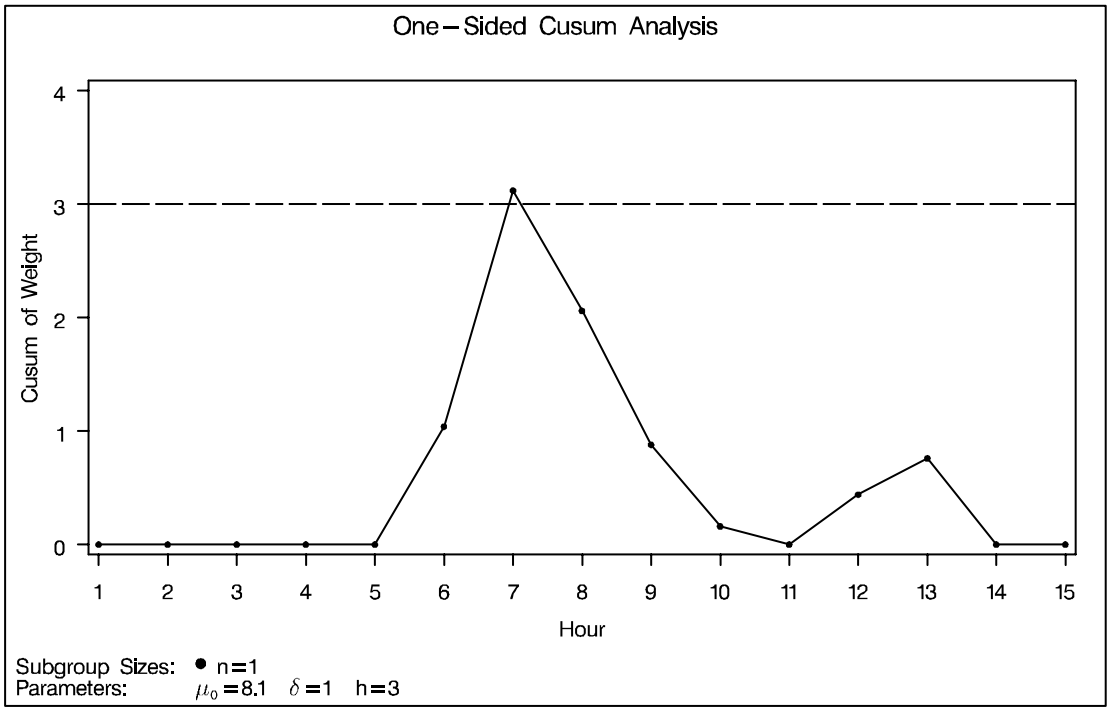


Figure 19.5. One-Sided Cusum Chart with Decision Interval

The cusum plotted at HOUR=*t* is

$$S_t = \max(0, S_{t-1} + (z_t - k))$$

where $S_0 = 0$, and z_t is the standardized deviation of the t^{th} measurement from the target.

$$z_t = \frac{x_t - \mu_0}{\sigma_0}$$

The cusum S_t is referred to as an *upper cumulative sum*. A shift is signaled at the seventh hour since S_7 exceeds h. For further details, see "One-Sided Cusum Schemes" on page 552.

The option TABLEALL requests the tables shown in Figure 19.6, Figure 19.7, and Figure 19.8. The table in Figure 19.6 summarizes the cusum scheme, and it confirms that an in-control ARL of 117.6 and an ARL of 6.4 at $\delta = 1$ are achieved with the specified h and k.

One-Sided Cusum Analysis

Cusum Parameters

Process Variable	weight (Cusum of Weight)
Subgroup Variable	hour (Hour)
Scheme	One-Sided
Target Mean (Mu0)	8.1
Sigma0	0.05
Delta	1
Nominal Sample Size	1
h	3
k	0.5
Average Run Length (Delta)	6.40390895
Average Run Length (0)	117.595692

Figure 19.6. Summary Table

The table in Figure 19.7 tabulates the information displayed in Figure 19.5.

Cumulative Sum Chart Summary for weight

hour	Subgroup Sample Size	Individual Value	Cusum	Decision Interval	Decision Interval Exceeded
1	1	8.0240000	0.0000000	3.0000	
2	1	7.9710000	0.0000000	3.0000	
3	1	8.1250000	0.0000000	3.0000	
4	1	8.1230000	0.0000000	3.0000	
5	1	8.0680000	0.0000000	3.0000	
6	1	8.1770000	1.0400000	3.0000	
7	1	8.2290000	3.1200000	3.0000	Upper
8	1	8.0720000	2.0600000	3.0000	
9	1	8.0660000	0.8800000	3.0000	
10	1	8.0890000	0.1600000	3.0000	
11	1	8.0580000	0.0000000	3.0000	
12	1	8.1470000	0.4400000	3.0000	
13	1	8.1410000	0.7600000	3.0000	
14	1	8.0470000	0.0000000	3.0000	
15	1	8.1250000	0.0000000	3.0000	

Figure 19.7. Tabulation of One-Sided Chart

The table in Figure 19.8 presents the computational form of the cusum scheme described by Lucas (1976).

Computational Cumulative Sum for weight

hour	Subgroup Sample Size	Individual Value	Upper Cusum	Number of Consecutive Upper Sums > 0
1	1	8.0240000	0.0000000	0
2	1	7.9710000	0.0000000	0
3	1	8.1250000	0.0000000	0
4	1	8.1230000	0.0000000	0
5	1	8.0680000	0.0000000	0
6	1	8.1770000	1.0400000	1
7	1	8.2290000	3.1200000	2
8	1	8.0720000	2.0600000	3
9	1	8.0660000	0.8800000	4
10	1	8.0890000	0.1600000	5
11	1	8.0580000	0.0000000	0
12	1	8.1470000	0.4400000	1
13	1	8.1410000	0.7600000	2
14	1	8.0470000	0.0000000	0
15	1	8.1250000	0.0000000	0

Figure 19.8. Computational Form of Cusum Scheme

Following the method of Lucas (1976), the process average at the out-of-control point (HOUR=7) can be estimated as

$$\begin{aligned}\mu_0 + \sigma_0(N_7 k + S_7)/(N_7\sqrt{n}) \\ = 8.10 + 0.05(2(0.5) + 3.12)/2 \\ = 8.203 \text{ ounces}\end{aligned}$$

where $S_7 = 3.12$ is the upper sum at HOUR=7, and $N_7 = 2$ is the number of successive positive upper sums at HOUR=7.

Saving Cusum Scheme Parameters

This example is a continuation of the previous example that illustrates how to save cusum scheme parameters in a data set specified with the OUTLIMITS= option. This enables you to apply the parameters to future data or to subsequently modify the parameters with a DATA step program.

See CUSONES1 in the SAS/QC Sample Library

```
title 'One-Sided Cusum Analysis';
proc cusum data=cans lineprinter;
   xchart weight*hour='*' /
      mu0    = 8.100     /* target mean for process     */
      sigma0 = 0.050     /* known standard deviation    */
      delta  = 1         /* shift to be detected        */
      h      = 3         /* cusum parameter h           */
      k      = 0.5       /* cusum parameter k           */
      scheme = onesided  /* one-sided decision interval */
 /*   nochart  */
      outlimits = cusparm
      ;
```

```
      label weight = 'Cusum of Weight';
   run;

   proc print data=cusparm;
   run;
```

The chart, shown in Figure 19.9, is similar to the one in Figure 19.5 but is created for output on a line printer since the LINEPRINTER option is included in the PROC CUSUM statement. *

```
                            One-Sided Cusum Analysis

     -----------------------------------------------------------------------
  4 +                                                                       |
C   |                                                                       |
u   |                                                                       |
s   |                                                                       |
u 3 +==============================*========================================|
m   |                             + +                                       |
    |                             +  ++                                     |
o   |                            +     +                                    |
f 2 +                            +      *                                   |
    |                           +        +                                  |
W   |                          +          ++                                |
e   |                          +            +                               |
i 1 +                         *              *                              |
g   |                        +                ++               ++*          |
h   |                      ++                   ++         +*++   ++        |
t   |                     +                       *++    ++         ++      |
  0 +*++++*++++*++++*++++*                           ++*+             *++++*|
     +----+----+----+----+----+----+----+----+----+----+----+----+----+----+
     1    2    3    4    5    6    7    8    9   10   11   12   13   14   15

                                     Hour

Subgroup Sizes:  * n=1
Parameters:      Mu0=8.1  Delta=1  h=3
```

Figure 19.9. One-Sided Cusum Scheme with Decision Interval

The OUTLIMITS= data set is listed in Figure 19.10.

```
          _            _          _                     _             _
          S            L          S         _           S     _       A
          U      _     I          C         D    _      T     A       R
    _     B      T     M          H      _  E    M      D     R       L
    V     G      Y     I          E      M  L    E      D     L       O
O   A     R      P     T _  _     M      U  T    A      E     I       U
b   R     P      E     N H  K     E      0  A    N      V     N       T
s   _     _      _     _ _  _     _      _  _    _      _     _       _

1 weight hour STANDARD 1 3 0.5 ONESIDED 8.1 1 8.09747 0.05 117.596 6.40391
```

Figure 19.10. Listing of the OUTLIMITS= Data Set CUSPARM

The data set contains one observation with the parameters for *process* WEIGHT. The variables _TYPE_, _H_, _K_, _MU0_, _DELTA_, and _STDDEV_ save the

*In Release 6.12 and previous releases of SAS/QC software, the keyword GRAPHICS was required in the PROC CUSUM statement to specify that the chart be created with a graphics device. In Version 7, you can specify the LINEPRINTER option to request line printer plots.

parameters specified with the options SCHEME=, H=, K=, MU0=, DELTA=, and SIGMA0=, respectively. The variable _MEAN_ saves an estimate of the process mean, and the variable _LIMITN_ saves the nominal sample size. The variables _ARLIN_ and _ARLOUT_ save the average run lengths for $\delta = 0$ and for $\delta = 1$.

The variables _VAR_ and _SUBGRP_ save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the value of _STDDEV_ is an estimate or a standard value.

For more information, see "OUTLIMITS= Data Set" on page 564.

Reading Cusum Scheme Parameters

This example shows how the cusum parameters saved in the previous example can be applied to new measurements saved in a data set named CANS2:

See CUSONES1 in the SAS/QC Sample Library

```
data cans2;
   length pump $ 8;
   label hour = 'Hour';
   input hour weight pump $ 8. ;
   datalines;
16  8.1765  Pump 3
17  8.0949  Pump 3
18  8.1393  Pump 3
19  8.1491  Pump 3
20  8.0473  Pump 1
21  8.1602  Pump 1
22  8.0633  Pump 1
23  8.0921  Pump 1
24  8.1573  Pump 1
25  8.1304  Pump 1
26  8.0979  Pump 1
27  8.2407  Pump 1
28  8.0730  Pump 1
29  8.0986  Pump 2
30  8.0785  Pump 2
31  8.2308  Pump 2
32  8.0986  Pump 2
33  8.0782  Pump 2
34  8.1435  Pump 2
35  8.0666  Pump 2
run;
```

The following statements create a one-sided cusum chart for the measurements in CANS2 using the parameters in CUSPARM:

```
title "One-Sided Cusum Analysis for New Data";
legend2 FRAME CFRAME=ligr CBORDER=black POSITION=center;
proc cusum data=cans2 limits=cusparm;
   xchart weight*hour ( pump );
   label weight = 'Cusum of Weight';
run;
```

The LIMITS= option in the PROC CUSUM statement specifies the data set containing preestablished cusum parameters.* The chart, shown in Figure 19.11, indicates that the process is in control. Levels of the variable PUMP (referred to as a *block-variable*) do not enter into the analysis but are displayed in a block legend across the top of the chart. See "Block Variable Legend Options" in Table 19.7 on page 540.

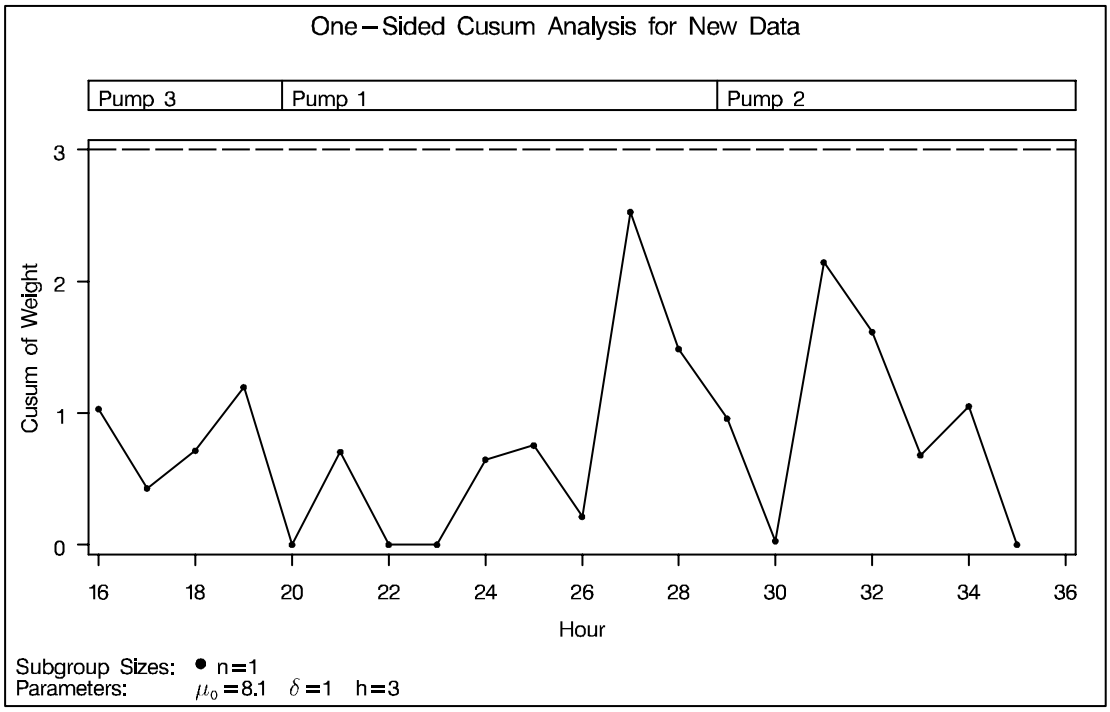


Figure 19.11. Cusum Chart with Decision Interval for New Data

In general, the parameters for a specified *process* and *subgroup-variable* are read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* (in this case, WEIGHT)
- the value of _SUBGRP_ matches the *subgroup-variable* name (in this case, HOUR)

If you are maintaining more than one set of cusum parameters for a particular *process*, you will find it convenient to include a special identifier variable named _INDEX_ in the LIMITS= data set. This must be a character variable of length 16. Then, if you specify READINDEX='*value*' in the XCHART statement, the parameters for a specified *process* and *subgroup-variable* are read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches *process*
- the value of _SUBGRP_ matches the *subgroup-variable* name
- the value of _INDEX_ matches *value*

*In Release 6.09 and in earlier releases, it is also necessary to specify the READLIMITS option or the READINDEX= option to read cusum parameters from a LIMITS= data set.

In this example, the LIMITS= data set was created in a previous run of the CUSUM procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 567 for details concerning the variables that you must provide.

Syntax

Specifying One-Sided Schemes

The basic syntax for a *one-sided (decision interval) scheme* using the XCHART statement is as follows:

XCHART *process***subgroup-variable* **/ SCHEME=ONESIDED MU0=***target* **DELTA=***shift* **H=***h* < *options* > **;**

The general form of this syntax is as follows:

XCHART (*processes***)***subgroup-variable* <**(** *block-variables* **)** > < =*symbol-variable* | ='*character*' > **/ SCHEME=ONESIDED MU0=***target* **DELTA=***shift* **H=***h* < *options* >**;**

Note that the options SCHEME=ONESIDED, MU0=, DELTA=, and H= are required unless their values are read from a LIMITS= data set.

Specifying Two-Sided Schemes

The basic syntax for a *two-sided (V-mask) scheme* is as follows:

XCHART *process***subgroup-variable* **/ MU0=***target* **DELTA=***shift* **ALPHA=***alpha*|**H=***h* < *options* > **;**

The general form of this syntax is as follows:

XCHART (*processes***)***subgroup-variable* <**(** *block-variables* **)**> < =*symbol-variable* | ='*character*' > **/ MU0=***target* **DELTA=***shift* **ALPHA=***alpha*|**H=***h* < *options* >**;**

Note that the options MU0=, DELTA=, and either ALPHA= or H= are required unless their values are read from a LIMITS= data set.

Components of the XCHART Statement

You can use any number of XCHART statements in the CUSUM procedure. The components of the XCHART statement are described as follows.

process
processes
identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC CUSUM statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating a V-Mask Cusum Chart from Raw Data" on page 522.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating a V-Mask Cusum Chart from Subgroup Summary Data" on page 525.

A *process* is required. If more than one *process* is specified, enclose the list in parentheses. The parameters specified in the XCHART statement are applied to all of the *processes*.*

subgroup-variable
is the variable that classifies the data into subgroups. The *subgroup-variable* is required. In the examples on pages 522 and 525, HOUR is the subgroup variable.

block-variables
are optionally specified variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See Figure 19.11 for an example.

symbol-variable
is an optionally specified variable whose levels (unique values) determine the plotting character or symbol marker used to plot the cusums.

- If you produce a chart on a line printer, an 'A' marks points corresponding to the first level of the *symbol-variable*, a 'B' marks points corresponding to the second level, and so on.
- If you produce a chart on a graphics device, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOLn statements.

character
specifies a plotting character for charts produced on line printers. See Figure 19.9 for an example.

options
specify optional cusum parameters, enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The "Summary of Options" section, which follows, lists all options by function.

*For this reason, it may be preferable to read distinct cusum parameters for each *process* from a LIMITS= data set.

Summary of Options

The following tables list the XCHART statement options by function. Options unique to the CUSUM procedure are listed in Table 19.1 to Table 19.5, and they are described in detail in "Dictionary of Special Options" on page 544. Options that are common to both the CUSUM and SHEWHART procedures are listed in Table 19.6 to Table 19.17. They are described in detail beginning on page 1851 in Chapter 53, "Dictionary of Options."

Table 19.1. Options for Specifying a One-Sided (Decision Interval) Cusum Scheme

DELTA=*value*	specifies shift to be detected as a multiple of standard error
H=*value*	specifies decision interval h ($h > 0$) as a multiple of standard error
HEADSTART=*value*	specifies headstart value S_0 as a multiple of standard error
K=*value*	specifies reference value k ($k > 0$)
LIMITN=*n*	specifies fixed nominal sample size for cusum scheme
LIMITN=VARYING	specifies that cusums are to be computed for all subgroups regardless of sample size
MU0=*value*	specifies target μ_0 for mean
NOREADLIMITS	specifies that cusum parameters are not to be read from LIMITS= data set (Release 6.10 and later releases)
READINDEX='*value*'	reads cusum scheme parameters from a LIMITS= data set
READLIMITS	specifies that cusum parameters are to be read from LIMITS= data set (Release 6.09 and earlier releases)
SCHEME=ONESIDED	specifies a one-sided scheme
SHIFT=*value*	specifies shift to be detected in data units
SIGMA0=*value*	specifies standard (known) value σ_0 for process standard deviation

Table 19.2. Options for Specifying a Two-Sided (V-Mask) Cusum Scheme

ALPHA=*value*	specifies probability of Type 1 error
BETA=*value*	specifies probability of Type 2 error
H=*value*	specifies vertical distance between V-mask origin and upper (or lower) arm
K=*value*	specifies slope of lower arm of V-mask
LIMITN=*n*	specifies fixed nominal sample size for cusum scheme
LIMITN=VARYING	specifies that cusums are to be computed for all subgroups regardless of sample size
NOREADLIMITS	specifies that cusum parameters are not to be read from LIMITS= data set (Release 6.10 and later releases)
READINDEX='*value*'	reads cusum scheme parameters from a LIMITS= data set
READLIMITS	specifies that cusum parameters are to be read from LIMITS= data set (Release 6.09 and earlier releases)
READSIGMAS	reads _SIGMAS_ instead of _ALPHA_ from LIMITS= data set when both variables are available
SIGMAS=*value*	specifies probability of Type 1 error as probability that standard normally distributed variable exceeds *value* in absolute value

Table 19.3. Options for Estimating Process Standard Deviation

SMETHOD=*keyword*	specifies method for estimating process standard deviation σ
TYPE=*keyword*	identifies whether _STDDEV_ in OUTLIMITS= data set is an estimate or standard, and specifies value of _TYPE_ in OUTLIMITS= data set

Table 19.4. Options for Displaying Decision Interval or V-Mask

CINFILL=*color*	specifies color for area under decision interval line or inside V-mask
CLIMITS=*color*	specifies color of decision interval line
CMASK=*color*	specifies color of V-mask outline
LLIMITS=*linetype*	specifies line type for decision interval
LMASK=*linetype*	specifies line type for V-mask arms
NOMASK	suppresses display of V-mask
ORIGIN=*value*|'*value*'	specifies value of *subgroup-variable* locating origin of V-mask
WLIMITS=*n*	specifies line width for decision interval
WMASK=*n*	specifies line width for V-mask

Table 19.5. Tabulation Options

TABLEALL	specifies the options TABLECHART, TABLECOMP, TABLEID, TABLEOUT, and TABLESUMMARY
TABLECHART	tabulates the information displayed in the cusum chart
TABLECOMP	tabulates the computational form of the cusum scheme as described by Lucas (1976) and Lucas and Crosier (1982)
TABLEID	augments TABLECHART and TABLECOMP tables with columns for ID variables
TABLEOUT	augments TABLECHART table with a column indicating if the decision interval or V-mask was exceeded
TABLESUMMARY	tabulates the parameters for the cusum scheme and the average run lengths corresponding to shifts of zero and δ

Note that specifying (EXCEPTIONS) after the option TABLECHART creates a table for exceptional points only.

Table 19.6. Axis and Axis Label Options

CAXIS=*color*	specifies color for axis lines and tick marks
CFRAME=*color*|(*color-list*)	specifies fill color(s) for frame for plot area
CTEXT=*color*	specifies color for tick mark values and axis labels
HAXIS=*values*|AXIS*n*	specifies tick mark values for horizontal axis
HEIGHT=*value*	specifies height of axis label and axis legend text
HMINOR=*n*	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=*value*	specifies length of offset at both ends of horizontal axis
INTSTART=*value*	specifies first major tick mark value for numeric horizontal axis
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=*n*	specifies thinning factor for tick mark labels on horizontal axis
SPLIT='*character*'	specifies splitting character for axis labels
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=*value-list*|*name*	specifies major tick mark values for vertical axis of cusum chart
VAXIS2=*value-list*|*name*	specifies major tick mark values for vertical axis of trend chart
VFORMAT=*format*	specifies format for tick mark labels on vertical axis of cusum chart
VFORMAT2=*format*	specifies format for tick mark labels on vertical axis of trend chart
VMINOR=*n*	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=*value*	specifies length of offset at both ends of vertical axis
WAXIS=*n*	specifies width of axis lines

Table 19.7. Block Variable Legend Options

BLOCKLABELPOS=*keyword*	specifies position of label for *block-variable* legend
BLOCKLABTYPE=*value*|*keyword*	specifies text size of *block-variable* legend
BLOCKPOS=*n*	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=*color-list*	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=*variable*|(*variables*)	specifies one or more variables whose values are colors for filling background of *block-variable* legend

Table 19.8. Reference Line Options

CHREF=*color*	specifies color for HREF= and HREF2= lines
CVREF=*color*	specifies color for VREF= and VREF2= lines
HREF=*values*|*SAS-data-set*	specifies reference lines perpendicular to horizontal axis on cusum chart
HREF2=*values*|*SAS-data-set*	specifies reference lines perpendicular to horizontal axis on trend chart
HREFCHAR='*character*'	specifies line character for HREF= and HREF2= lines
HREFDATA=*SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on cusum chart
HREF2DATA=*SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFLABELS='*label1*'...'*labeln*'	specifies labels for HREF= lines
HREF2LABELS='*label1*'...'*labeln*'	specifies labels for HREF2= lines
HREFLABPOS=*n*	specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=*linetype*	specifies line type for HREF= and HREF2= lines
LVREF=*linetype*	specifies line type for VREF= and VREF2= lines
NOBYREF	specifies that reference line information in a data set is to be applied uniformly to charts created for all BY groups
VREF=*values*|*SAS-data-set*	specifies reference lines perpendicular to vertical axis on cusum chart
VREF2=*values*|*SAS-data-set*	specifies reference lines perpendicular to vertical axis on trend chart
VREFCHAR='*character*'	specifies line character for VREF= and VREF2= lines
VREFLABELS='*label1*'...'*labeln*'	specifies labels for VREF= lines
VREF2LABELS='*label1*'...'*labeln*'	specifies labels for VREF2= lines
VREFLABPOS=*n*	specifies position of VREFLABELS= and VREF2LABELS= labels

Table 19.9. Options for Plotting and Labeling Points

ALLLABEL=VALUE|(*variable*)	labels every point on cusum chart
ALLLABEL2=VALUE|(*variable*)	labels every point on trend chart
CCONNECT=*color*	specifies color for line segments that connect points on chart
CFRAMELAB=*color*	specifies fill color for frame around labeled points
CLABEL=*color*	specifies color for labels
CONNECTCHAR='*character*'	specifies character used to form line segments that connect points on chart
COUT=*color*	specifies color for connecting line segments that lie above the decision interval or outside the V-mask
COUTFILL=*color*	specifies color for areas between connected points and decision interval or V-mask
LABELANGLE=*angle*	specifies angle at which labels are drawn
LABELFONT=*font*	specifies software font for labels
LABELHEIGHT=*value*	specifies height of labels
NOCONNECT	suppresses line segments that connect points on chart
NOTRENDCONNECT	suppresses line segments that connect points on trend chart
OUTLABEL=VALUE|(*variable*)	labels points exceeding decision interval on one-sided chart
SYMBOLCHARS='*characters*'	specifies characters indicating *symbol-variable*
SYMBOLLEGEND=NONE|*name*	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=*keyword*	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL	turns point labels so that they are strung out vertically

Table 19.10. Phase Options

CPHASELEG=*color*	specifies text color for *phase* legend
OUTPHASE='*string*'	specifies value of _PHASE_ in OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=*value*|*keyword*	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASEREF	delineates *phases* with vertical reference lines
READPHASES= ALL|'*label1*'...'*labeln*'	specifies *phases* to be read from input data set

Table 19.11. Grid Options

CGRID=*color*	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to chart
LENDGRID=*linetype*	specifies line type for grid requested with the ENDGRID option
LGRID=*linetype*	specifies line type for grid requested with the GRID option
WGRID=*n*	specifies width of grid lines

Table 19.12. Graphical Enhancement Options

ANNOTATE= *SAS-data-set*	specifies annotate data set that adds features to cusum chart
ANNOTATE2= *SAS-data-set*	specifies annotate data set that adds features to trend chart
DESCRIPTION='*string*'	specifies string that appears in the description field of PROC GREPLAY master menu for cusum chart
FONT=*font*	specifies software font for labels and legends on charts
NAME='*string*'	specifies name that appears in the name field of the PROC GREPLAY master menu for cusum chart
PAGENUM='*string*'	specifies the form of the label used in pagination
PAGENUMPOS= *keyword*	specifies the position of the page number requested with the PAGENUM= option
WTREND=*n*	specifies width of line segments connecting points on trend chart

Table 19.13. Input Data Set Options

MISSBREAK	specifies that observations with missing values are not to be processed

Table 19.14. Output Data Set Options

OUTHISTORY= *SAS-data-set*	creates output data set containing subgroup summary statistics and cusums
OUTINDEX='*string*'	specifies value of _INDEX_ in OUTLIMITS= data set
OUTLIMITS= *SAS-data-set*	creates output data set containing cusum parameters
OUTPHASE='*string*'	specifies value of _PHASE_ in OUTHISTORY= data set or OUTTABLE= data set
OUTTABLE= *SAS-data-set*	creates output data set containing subgroup summary statistics, cusums, and decision interval or V-mask values

Table 19.15. Plot Layout Options

ALLN	plots summary statistics for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates cusum chart only when exceptions occur
INTERVAL=*keyword*	specifies natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=*n*	specifies maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of cusum chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes and cusum parameters
NPANELPOS=*n*	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=*n*	specifies number of pages or screens to be used to display chart
TRENDVAR=*variable*| (*variable-list*)	specifies list of trend variables
YPCT1=*value*	specifies length of vertical axis on cusum chart as a percentage of sum of lengths of vertical axes for cusum and trend charts

Table 19.16. Options for Interactive Control Charts

HTML=(*variable*)	specifies a variable whose values are URLs to be associated with subgroups
HTML2=(*variable*)	specifies variable whose values are URLs to be associated with points on trend chart
HTML_LEGEND= (*variable*)	specifies a variable whose values are URLs to be associated with symbols in the symbol legend
WEBOUT=*SAS-data-set*	creates an OUTTABLE= data set with additional graphics coordinate data

Table 19.17. Star Options

CSTARCIRCLES=*color*	specifies color for circles requested by the STARCIRCLES= option
CSTARFILL=*color*|*(variable)*	specifies color for filling stars
CSTAROUT=*color*	specifies color for outlines of stars that exceed inner or outer circles
CSTARS=*color*|*(variable)*	specifies color for outlines of stars
LABELFONT=*font*	specifies font for labels added with the STARLABEL= option
LSTARCIRCLES=*linetypes*	specifies line types of circles requested by the STARCIRCLES= option
LSTARS=*linetype*|*(variable)*	specifies line types of outlines of stars requested by the STARVERTICES= option
STARBDRADIUS=*value*	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=*value-list*	specifies reference circles for stars
STARINRADIUS=*value*	specifies inner radius of stars
STARLABEL=*keyword*	specifies vertices to be labeled
STARLEGEND=*keyword*	specifies style of legend for star vertices
STARLEGENDLAB=*'label'*	specifies label for STARLEGEND= legend
STAROUTRADIUS=*value*	specifies outer radius of stars
STARSPEC=*value*|*SAS-data-set*	specifies method used to standardize vertex variables
STARSTART=*value*	specifies angle for first vertex
STARTYPE=*keyword*	specifies graphical style of star
STARVERTICES=*variable*|*(variables)*	superimposes star at each point on chart
WSTARCIRCLES=*n*	specifies width of circles requested by the STARCIRCLES= option
WSTARS=*n*	specifies width of stars requested by the STARVERTICES= option

Dictionary of Special Options

The marginal notes *Graphics* and *Line Printer* identify options that apply to graphics devices and line printers, respectively.

ALPHA=*value*

specifies the probability α of incorrectly deciding that a shift has occurred when the process mean is equal to the target mean. This is known as the probability of a Type

1 error. The *value* must be between zero and one, and it is typically set at 0.05 or 0.10. If you specify the ALPHA= option, the error probability approach is used to determine the V-mask. For details, see "Defining the V-Mask for a Two-Sided Cusum Scheme" on page 555.

The ALPHA= option is applicable only with two-sided cusum schemes. As an alternative to the ALPHA= *value*, you can specify the percentile $z_{1-\alpha/2}$ from a standard normal distribution with the SIGMAS= option. As a second alternative, you can specify the geometric parameter h for the V-mask (in standard error units) with the H= option.

In addition to the ALPHA= option, you can optionally specify the probability of a Type 2 error with the BETA= option.

BETA=*value*
specifies the probability β of failing to discover that the specified shift has occurred. This is known as the probability of a Type 2 error. The *value* must be between zero and one. The BETA= option is used in conjunction with either the ALPHA= option or the SIGMAS= option.

The interpretation of β is based on the analogy between cusum charts and sequential probability ratio tests, and it is inexact since the cusum chart does not provide an acceptance region. Refer to Johnson (1961) and van Dobben de Bruyn (1968) for further details.

CINFILL=*color*
specifies the color for the area under the decision interval or inside the V-mask arms. By default, these areas are not filled with a color. See also the COUTFILL= option. *Graphics*

CLIMITS=*color*
specifies the color for the decision interval line. The default color is the first color in the device color list. *Graphics*

CMASK=*color*
specifies the color for the V-mask arms. The default color is the first color in the device color list. *Graphics*

DATAUNITS
computes cumulative sums without standardizing the subgroup means or individual measurements. As a result, the vertical axis of the cusum chart is scaled in the same units as the data.

The DATAUNITS option requires constant subgroup sample sizes. If your data do not have constant subgroup sample sizes, you need to specify a constant nominal sample size n for the V-mask or decision interval with the LIMITN= option or with the variable _LIMITN_ in the LIMITS= data set.

DELTA=*value*
specifies the absolute value of the smallest shift to be detected as a multiple δ of the process standard deviation σ or the standard error $\sigma_{\bar{X}}$, depending on whether δ is viewed as a shift in the population mean or a shift in the sampling distribution of the subgroup mean $\bar{X}$, respectively.

If you specify SCHEME=ONESIDED (see the SCHEME= option later in this list) and the *value* is positive, a shift above the process mean is to be detected, whereas if the *value* is negative, a shift below the process mean is to be detected.

As an alternative to specifying the DELTA= option, you can specify the shift in the same units as the data with the SHIFT= option.

H=*value*

specifies the decision interval h for a one-sided cusum scheme. This type of scheme is completely specified by the parameters h and k (see the K= option later in this list). You can also specify the H= option as an alternative to the ALPHA= or SIGMAS= options for a two-sided cusum scheme with a V-mask. In this case, the H= option specifies the vertical distance h between the origin for the V-mask and the upper or lower arm of the V-mask. In either case, the H= *value* must be positive and must be expressed as a multiple of standard error.

You can use a table of average run lengths to choose h (this is typically between zero and 10). See pages 559 and 560.

HEADSTART=*value*
HSTART=*value*

specifies a headstart value S_0 for a one-sided cusum scheme. The value must be expressed as a multiple of standard error. See "Headstart Values" on page 553, and refer to Lucas and Crosier (1982), Ryan (1989), and Montgomery (1996).

K=*value*

specifies the reference value k for a one-sided (decision interval) cusum scheme. This type of scheme is completely specified by the parameters k and h (see the H= option earlier in this list). You can also specify the K= and H= options as geometric parameters for a two-sided cusum scheme with a V-mask . In this case, the K= option specifies the slope of the lower arm of the V-mask, and the K= and H= options together are alternatives to the error probability options ALPHA=, SIGMAS=, and BETA=. In either case, the K= *value* must be positive and must be expressed as a multiple of standard error.

You can use a table of average run lengths to choose k and h (k is typically between zero and two). See pages 559 and 560.

For a one-sided scheme, the default K= *value* is $\delta/2$, which is referred to as the *central reference value*. For a two-sided scheme where the V-mask is specified geometrically with the H= option, the default K= *value* is $\delta/2$. If, however, the V-mask is specified by an error probability with the ALPHA= option, then the K= option should not be specified.

CAUTION: The interpretation of the K= *value* depends on the *subgroup-variable* and the interval between subgroups that is specified with the INTERVAL= option. For a two-sided scheme, the *value* is the increase in the lower V-mask arm per unit change on the subgroup axis, so the *value* depends on how the *subgroup-variable* is scaled.

- If integer values are assigned to the *subgroup-variable*, then a unit change is defined as one.
- If the *subgroup-variable* has character values, then a unit change is defined as the increment between adjacent values of the *subgroup-variable*.
- If the *subgroup-variable* is numeric and is formatted with a SAS date or time format, then a unit change is defined as the default value for the INTERVAL= option. For example, if a DATE7. format is associated with the *subgroup-variable*, then a unit change is defined as one day.

You can use the INTERVAL= option to modify the definition of a unit change. For example, if a DATE7. format is associated with the *subgroup-variable* but subgroups are collected hourly, then INTERVAL=HOUR defines a unit change as one hour rather than one day.

LIMITN=*n*
LIMITN=VARYING
specifies either a fixed or varying nominal sample size for the control limits. If you specify LIMITN=n, cusums are calculated and displayed only for those subgroups with a sample size equal to n, although you can specify the ALLN option to force all cusums to be plotted. If you specify LIMITN=VARYING, cusums are calculated and displayed for all subgroups, regardless of sample size.

LLIMITS=*linetype*
specifies the line type for the decision interval. The default is 4 (a dashed line). *Graphics*

LMASK=*linetype*
specifies the line type for the V-mask arms. The default is 1 (a solid line). *Graphics*

MU0=*value*
specifies the target mean μ_0 for the process. The target mean must be scaled in the same units as the data.

NOARL
suppresses calculation of average run lengths. By default, this calculation is performed if you specify the TABLESUMMARY option or an OUTLIMITS= data set.

NOMASK
suppresses the display of the V-mask on charts for two-sided schemes. This option does not affect computations of cusums or V-mask parameters.

NOREADLIMITS
specifies that the cusum scheme parameters for each *process* listed in the chart statement are *not* to be read from the LIMITS= data set specified in the PROC CUSUM statement. The NOREADLIMITS option is available only in Release 6.10 and later releases. See the READLIMITS option later in this list.

ORIGIN=*value*
specifies the origin of the V-mask, which is defined as the horizontal coordinate of the right edge of the V-mask. If a date, time, or datetime format is associated with the *subgroup-variable*, you must specify the *value* as a date, time, or datetime constant, respectively. If the subgroup variable is character, you must specify the *value* as

a quoted string. The default *value* is the last (most recent) value of the *subgroup-variable*.

Note that estimates for the process mean and standard deviation are calculated only from subgroups up to and including the origin subgroup.

READINDEX=*'value'*
reads cusum scheme parameters from a LIMITS= data set (specified in the PROC CUSUM statement) for each *process* listed in the chart statement. The t^{th} set of control limits for a particular *process* is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches *process*
- the value of _SUBGRP_ matches the *subgroup-variable*
- the value of _INDEX_ matches *value*

The *value* can be up to 16 characters and must be enclosed in quotes.

READLIMITS
specifies that cusum scheme parameters are to be read from a LIMITS= data set specified in the PROC CUSUM statement. The parameters for a particular *process* are read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches *process*
- the value of _SUBGRP_ matches the *subgroup variable*

The use of the READLIMITS option depends on which release of SAS/QC software you are using.

- **In Release 6.10 and later releases, the READLIMITS option is not necessary.** To read cusum scheme parameters as described previously, you simply specify a LIMITS= data set. However, even though the READLIMITS option is redundant, it continues to function as in earlier releases.
- **In Release 6.09 and earlier releases, you must specify the READLIMITS option to read cusum scheme parameters as described previously.** If you specify a LIMITS= data set without specifying the READLIMITS option (or the READINDEX= option), the cusum scheme parameters are computed from the data.

READSIGMAS
specifies that the variable _SIGMAS_ (instead of _ALPHA_) is to be read from a LIMITS= data set that contains both variables. The variables _SIGMAS_ and _ALPHA_ provide the same parameters as the SIGMAS= and ALPHA= options. By default, _ALPHA_ is read from the LIMITS= data set.

SCHEME=ONESIDED
SCHEME=TWOSIDED
indicates whether the cusum scheme is a one-sided (decision interval) scheme or a two-sided scheme with a V-mask. By default, SCHEME=TWOSIDED.

SHIFT=*value*
specifies the shift to be detected in the same units as the data. The *value* is interpreted as the shift in the mean of the sampling distribution of the subgroup mean. The SHIFT= option is an alternative to the DELTA= option. To specify the SHIFT= option, one of the following must be true:

- The subgroup sample sizes are constant.
- A constant nominal sample size n is provided for the cusum scheme with the LIMITN= option or the _LIMITN_ variable in a LIMITS= data set.

The relationship between the SHIFT= *value* (denoted by Δ) and the DELTA= value (denoted by δ) is $\delta = \Delta/(\sigma/\sqrt{n})$, where σ is the process standard deviation.

SIGMA0=*value*
specifies a known standard deviation σ_0 for the process standard deviation σ. The *value* must be positive. By default, PROC CUSUM estimates σ from the data using the formulas given in "Methods for Estimating the Standard Deviation" on page 561. You can use the variable _STDDEV_ in a LIMITS= data set as an alternative to the SIGMA0= option.

SIGMAS=*value*
specifies the probability α of false detection for a two-sided cusum scheme with a V-mask as the probability that the absolute value of a standard normally distributed variable is greater than the *value*. For example, SIGMAS=3 corresponds to the probability α =0.0027. The *value* must be positive. The SIGMAS= option is an alternative to the ALPHA= and H= options, and only one of these three options can be specified.

The SIGMAS= option is useful for defining cusum charts that correspond to Shewhart charts whose control limits are defined with the same *value* as the multiple of σ. Refer to Johnson and Leone (1962, 1974).

SMETHOD=NOWEIGHT | MVLUE | RMSDF
specifies a method for estimating the process standard deviation from subgroup observations, σ, as summarized by the following table.

Keyword	Method for Estimating Standard Deviation
NOWEIGHT	estimates σ as an unweighted average of unbiased subgroup estimates of σ
MVLUE	calculates a minimum variance linear unbiased estimate for σ
RMSDF	calculates a root-mean square estimate for σ

For formulas, see "Methods for Estimating the Standard Deviation" on page 561.

TABLEALL
requests all the tables specified by the options TABLECHART, TABLECOMP, TABLEID, TABLEOUT, and TABLESUMMARY.

TABLECHART <**(EXCEPTIONS)**>
creates a table of the subgroup variable, the subgroup sample sizes, the subgroup means, the cumulative sums, and the decision interval or V-mask limits. A table is produced for each *process* specified in the XCHART statement. The keyword EXCEPTIONS (enclosed in parentheses) is optional and restricts the tabulation to those subgroups for which the decision interval or V-mask values are exceeded.

TABLECOMP
tabulates the computational form of the cusum scheme as described by Lucas (1976) and Lucas and Crosier (1982). Upper or lower cumulative sums (or both) are tabulated for each *process* given in the XCHART statement. See "Formulas for Cumulative Sums" on page 552 for more information.

TABLEID
augments the tables specified by the TABLECHART and TABLECOMP options with a column for each of the ID variables.

TABLEOUT
augments the table specified by the TABLECHART option with a column indicating whether the decision interval or V-mask values are exceeded.

TABLESUMMARY
produces a table that summarizes the cusum scheme. The table lists the parameters of the scheme and the average run lengths corresponding to shifts of zero and δ. The average run lengths are computed using the method of Goel and Wu (1971). A table is produced for each *process*. You can save the summary in a data set by specifying the OUTLIMITS= option. See "OUTLIMITS= Data Set" on page 564 for details.

TYPE=ESTIMATE
TYPE=STANDARD
specifies the value of _TYPE_ in an OUTLIMITS= data set. The variable _TYPE_ indicates whether the variable _STDDEV_ in the OUTLIMITS= data set represents an estimate or a standard (known) value. The default is `STANDARD` if the SIGMA0= option is specified; otherwise, the default is `ESTIMATE`.

WLIMITS=*linetype*
Graphics
specifies the width (in pixels) of the decision interval line. The default width is 1.

WMASK=*linetype*
Graphics
specifies the width (in pixels) of the V-mask arms. The default width is 1.

Details

Basic Notation for Cusum Charts

The following notation is used in this chapter:

μ denotes the mean of the population, also referred to as the *process mean* or the *process level*.

μ_0 denotes the target mean (goal) for the population. Goel and Wu (1971) refer to μ_0 as the "acceptable quality level" and use the symbol μ_a instead. The symbol $\bar{X}_0$ is used for μ_0 in *Glossary and Tables for Statistical Quality Control*. You can provide μ_0 with the MU0= option or with the variable _MU0_ in a LIMITS= data set.

σ denotes the population standard deviation. You can provide σ with the variable _STDDEV_ in a LIMITS= data set (where _TYPE_=`STANDARD`).

σ_0 denotes a known standard deviation. You can provide σ_0 with the SIGMA0= option or the variable _STDDEV_ in a LIMITS= data set.

$\hat{\sigma}$ denotes an estimate of σ. You can provide $\hat{\sigma}$ with the SIGMA0= option or the variable _STDDEV_ in a LIMITS= data set. To identify this value as an estimate, specify TYPE=ESTIMATE or assign the value `ESTIMATE` to the variable _TYPE_ in a LIMITS= data set.

n denotes the nominal sample size for the cusum scheme. You can provide n with the LIMITN= option or the variable _LIMITN_ in a LIMITS= data set.

δ denotes the shift in μ to be detected, expressed as a multiple of the standard deviation. You can provide δ with the DELTA= option or the variable _DELTA_ in a LIMITS= data set.

Δ denotes the shift in μ to be detected, expressed in data units. If the sample size n is constant across subgroups, then $\Delta = \delta\sigma_{\bar{X}} = (\delta\sigma)/\sqrt{n}$.
Some authors use the symbol D instead of Δ; for example, refer to Johnson and Leone (1962, 1974) and Wadsworth and others (1986). You can provide Δ with the SHIFT= option. Although it may be more natural to specify the shift in data units, it is preferable to specify the shift as δ, since this generalizes to data with unequal subgroup sample sizes.

Formulas for Cumulative Sums

One-Sided Cusum Schemes

Positive Shifts

If the shift δ to be detected is positive, the cusum computed for the t^{th} subgroup is

$$S_t = \max(0, S_{t-1} + (z_t - k))$$

for t=1, 2, . . . , n, where S_0=0, z_t is defined as for two-sided schemes, and the parameter k, termed the *reference value*, is positive. The cusum S_t is referred to as an *upper cumulative sum*. Since S_t can be written as

$$\max\left(0, S_{t-1} + \frac{\bar{X}_i - (\mu_0 + k\sigma_{\bar{X}_i})}{\sigma_{\bar{X}_i}}\right)$$

the sequence S_t cumulates deviations in the subgroup means greater than k standard errors from μ_0. If S_t exceeds a positive value h (referred to as the *decision interval*), a shift or out-of-control condition is signaled. This formulation follows that of Lucas (1976), Lucas and Crosier (1982), and Montgomery (1996).

Negative Shifts

If the shift δ to be detected is negative, the cusum computed for the t^{th} subgroup is

$$S_t = \max(0, S_{t-1} - (z_t + k))$$

for t=1, 2, . . . , n, where S_0=0, z_t is defined as for two-sided cusum schemes, and the parameter k, termed the *reference value*, is positive. The cusum S_t is referred to as a *lower cumulative sum*. Since S_t can be written as

$$\max\left(0, S_{t-1} - \frac{\bar{X}_i - (\mu_0 - k\sigma_{\bar{X}_i})}{\sigma_{\bar{X}_i}}\right)$$

the sequence S_t cumulates the absolute value of deviations in the subgroup means less than k standard errors from μ_0. If S_t exceeds a positive value h (referred to as the *decision interval*), a shift or out-of-control condition is signaled.

This formulation follows that of Lucas (1976), Lucas and Crosier (1982), and Montgomery (1996). Note that S_t is always positive and h is always positive, regardless of whether δ is positive or negative. For schemes designed to detect a negative shift, some authors, including van Dobben de Bruyn (1968) and Wadsworth and others (1986), define a reflected version of S_t for which a shift is signaled when S_t is less than a negative limit.

Headstart Values

Lucas and Crosier (1982) describe the properties of a fast initial response (FIR) feature for cusum schemes in which the initial cusum S_0 is set to a "headstart" value. Average run length calculations given by Lucas and Crosier (1982) show that the FIR feature has little effect when the process is in control and that it leads to a faster response to an initial out-of-control condition than a standard cusum scheme. You can provide headstart value S_0 with the HEADSTART= option or the variable _HSTART_ in a LIMITS= data set.

Constant Sample Sizes

When the subgroup sample sizes are constant (=n), it may be preferable to compute cusums that are scaled in the same units as the data. Refer to Montgomery (1996) and Wadsworth and others (1986). To request this, specify the DATAUNITS option. Cusums are then computed as

$$S_t = \max(0, S_{t-1} + (\bar{X}_i - (\mu_0 + k\sigma/\sqrt{n})))$$

for δ >0 and the equation

$$S_t = \max(0, S_{t-1} - (\bar{X}_i - (\mu_0 - k\sigma/\sqrt{n})))$$

for $\delta < 0$. In either case, a shift is signaled if S_t exceeds $h' = h\sigma/\sqrt{n}$. Wadsworth and others (1986) use the symbol H for h'.

If the subgroup sample sizes are not constant, you can specify a constant nominal sample size n with the LIMITN= option or the variable _LIMITN_ in a LIMITS= data set. In this case, only those subgroups with sample size n are analyzed unless you also specify the option ALLN. You can further specify the option NMARKERS to request special symbol markers for points corresponding to sample sizes not equal to n.

Two-Sided Cusum Schemes

If the cusum scheme is two-sided, the cumulative sum S_t plotted for the t^{th} subgroup is

$$S_t = S_{t-1} + z_t$$

for t=1, 2, . . . , n. Here S_0=0, and the term z_t is calculated as

$$z_t = (\bar{X}_t - \mu_0)/(\sigma/\sqrt{n_t})$$

where $\bar{X}_t$ is the t^{th} subgroup average, and n_t is the t^{th} subgroup sample size. If the subgroup samples consist of individual measurements x_t, the term z_t simplifies to

$$z_t = (x_t - \mu_0)/\sigma$$

Since the first equation can be rewritten as

$$S_t = \sum_{i=1}^{t} z_i = \sum_{i=1}^{t} (\bar{X}_i - \mu_0)/\sigma_{\bar{X}_i}$$

the sequence S_t cumulates standardized deviations of the subgroup averages from the target mean μ_0.

In many applications, the subgroup sample sizes n_i are constant ($n_i = n$), and the equation for S_t can be simplified.

$$S_t = (1/\sigma_{\bar{X}}) \sum_{i=1}^{t} (\bar{X}_i - \mu_0) = (\sqrt{n}/\sigma) \sum_{i=1}^{t} (\bar{X}_i - \mu_0)$$

In some applications, it may be preferable to compute S_t as

$$S_t = \sum_{i=1}^{t} (\bar{X}_i - \mu_0)$$

which is scaled in the same units as the data. Refer to Montgomery (1996), Wadsworth and others (1986), and *ASQC Glossary and Tables for Statistical Quality Control*. If the subgroup sample sizes are constant (= n) and if you specify the DATAUNITS option in the XCHART statement, the CUSUM procedure computes cusums using the final equation above. In this case, the procedure rescales the V-mask parameters h and k to $h' = h\sigma/\sqrt{n}$ and $k' = k\sigma/\sqrt{n}$, respectively. Wadsworth and others (1986) use the symbols F for k' and H for h'.

If the subgroup sample sizes are not constant, you can specify a constant nominal sample size n with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set. In this case, only those subgroups with sample size n are analyzed unless you also specify the option ALLN. You can further specify the option NMARKERS to request special symbol markers for points corresponding to sample sizes not equal to n.

If the process is in control and the mean μ is at or near the target μ_0, the points will not exhibit a trend since positive and negative displacements from μ_0 tend to cancel each other. If μ shifts in the positive direction, the points exhibit an upward trend, and if μ shifts in the negative direction, the points exhibit a downward trend.

Defining the Decision Interval for a One-Sided Cusum Scheme

The height of the decision interval is h, expressed as a multiple of the standard error of the subgroup mean. You can specify h with the H= option in the XCHART statement or with the variable _H_ in a LIMITS= data set. The decision interval is displayed as a horizontal line on the cusum chart, as illustrated in Figure 19.12.

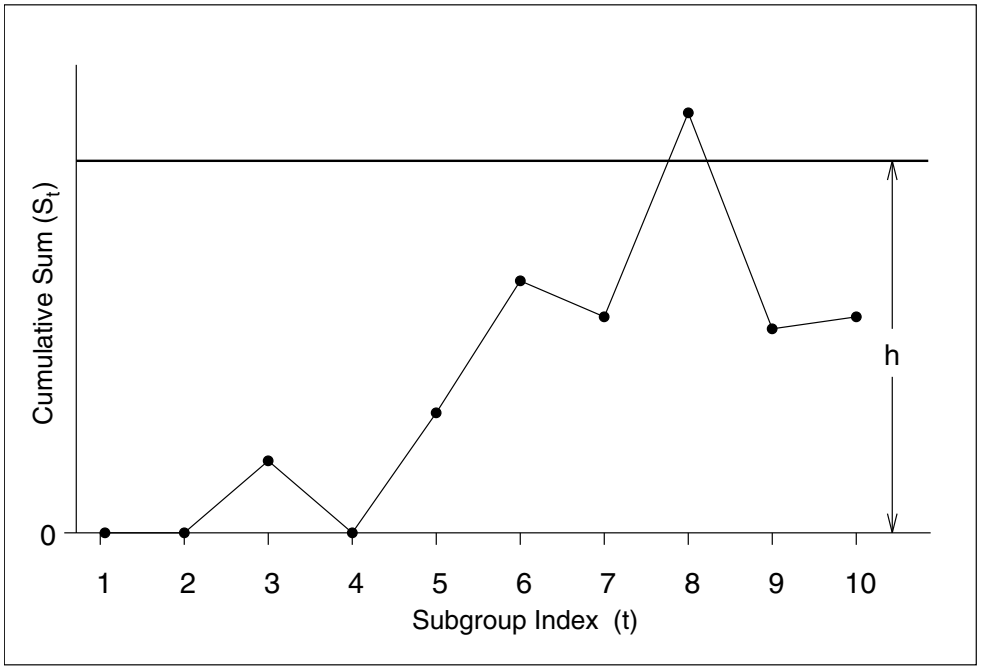


Figure 19.12. Decision Interval

Interpreting One-Sided Cusum Charts

A shift or out-of-control condition is signaled at time t if the cusum S_t plotted at time t exceeds the decision interval line.

Defining the V-Mask for a Two-Sided Cusum Scheme

The dimensions of the V-mask can be specified using two distinct sets of two parameters.

- θ, defined as half of the angle formed by the V-mask arms, and d, the distance between the origin and the vertex, as shown in Figure 19.13. This parameterization is used by many authors, including Johnson and Leone (1962, 1974) and Montgomery (1996).
- h, the vertical distance between the origin and the upper (or lower) V-mask arm, and k, the rise (drop) in the lower (upper) arm corresponding to an interval of one subgroup unit on the horizontal axis. You can specify the definition of an interval with the INTERVAL= option. This parameterization is used by Lucas (1976) and Wadsworth and others (1986). Lucas (1976) uses the symbols h^* for h and k^* for k, and Wadsworth and others (1986) use the symbol f in place of k.

The two parameterizations are related by the equations

$$\theta = \arctan(k/a)$$

$$d = h/k$$

where the aspect ratio a is the number of units on the vertical axis corresponding to one unit on the horizontal axis. The CUSUM procedure uses the h and k parameterization because it eliminates the need for working with aspect ratios, which are dependent on the graphics device. Furthermore, h and k are also useful for average run length computations and for parameterizing one-sided cusum schemes.

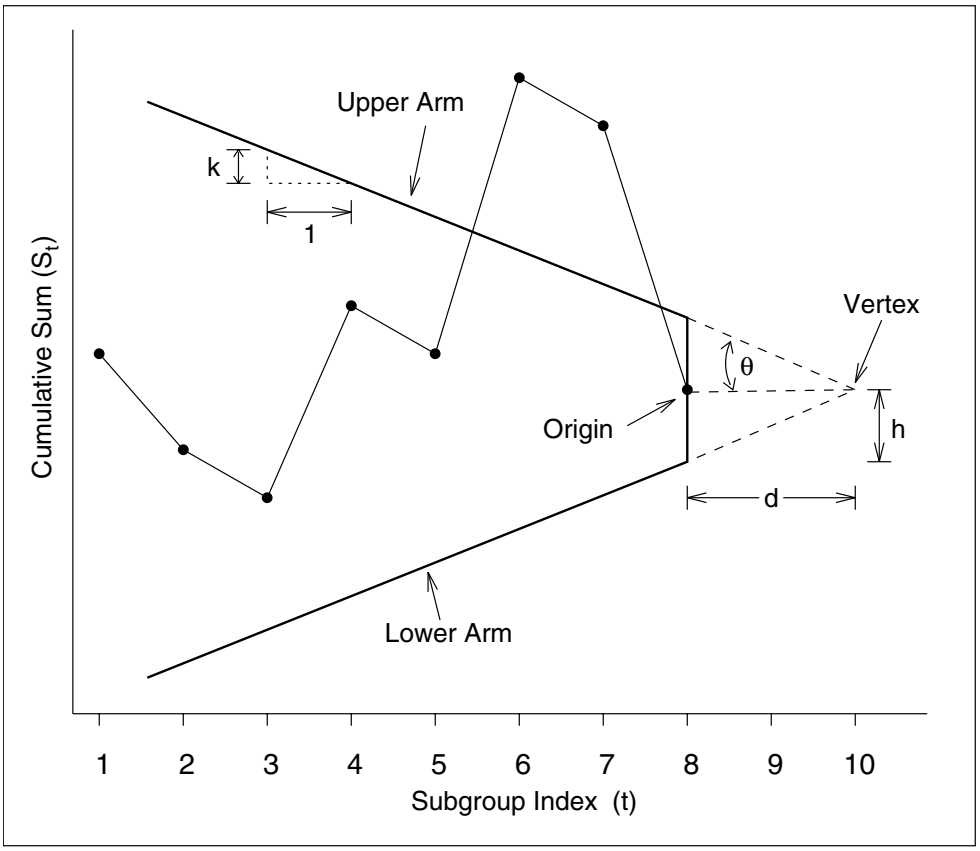


Figure 19.13. V-Mask Parameters

You can specify the V-mask in two ways:

- geometrically, by providing h and k (or simply h) with the H= and K= options or with the variables _H_ and _K_ in a LIMITS= data set
- in terms of error probabilities, by providing α and β (or simply α) with the ALPHA= and BETA= options or with the variables _ALPHA_ and _BETA_ in a LIMITS= data set. The SIGMAS= option is an alternative to the ALPHA= option, and the variable _SIGMAS_ is an alternative to the variable _ALPHA_ (if the READSIGMAS option is specified).

If you provide α and β, h and k are computed using the formulas

$$h = |\delta|^{-1} \log((1-\beta)/(\alpha/2))$$

$$k = |\delta|/2$$

If you provide α but not β, h and k are computed using the formulas

$$h = -|\delta|^{-1} \log(\alpha/2)$$

$$k = |\delta|/2$$

In the preceding equations, the error probability α is divided by two because two-sided deviations from the target mean are detected. Refer to Johnson and Leone (1962, 1974).

Interpreting Two-Sided Cusum Charts

The origin of the V-mask is located at the most recently plotted point, as illustrated in Figure 19.13. As additional data are collected and the cumulative sum sequence is updated, the origin is relocated at the newest point. A shift or out-of-control condition is signaled at time t if one or more of the points plotted up to time t cross an arm of the V-mask. An upward shift is signaled by points crossing the lower arm, and a downward shift is signaled by points crossing the upper arm. The time at which the shift occurred corresponds to the time at which a distinct change is observed in the slope of the plotted points.

Designing a Cusum Scheme

There are three main methods for designing a cusum scheme: the *average run length (ARL) approach*, the *error probability approach*, and the *economic design approach*.

Average Run Length (ARL) Approach

With the ARL approach, the parameters h and k are chosen to yield desired average run lengths when the process is operating at the target mean and when a shift of magnitude δ has occurred. The average run length is the expected number of samples taken before an out-of-control condition is signaled. Ideally, the ARL should be long when $\mu = \mu_0$ and short when μ shifts away from μ_0.

The ARL method typically involves the use of a table or nomogram. Refer to Kemp (1961), van Dobben de Bruyn (1968), Goel and Wu (1971), Duncan (1974), Lucas (1976), Montgomery (1996), and Wadsworth and others (1986).

For one-sided charts, average run lengths are tabulated as a function of h, k, and δ in Table 19.18 on page 559. No headstart is assumed in this table. For two-sided charts, average run lengths are tabulated as a function of h, k, and δ in Table 19.19 on page 560, which is formatted similarly to Table 2 given by Lucas (1976).

The ARLs in Table 19.18 and Table 19.19 were calculated with the DATA step function CUSUMARL, which is described on page 2099. This function uses the method of Goel and Wu (1971). You can use this function to generate more detailed, interpolated versions of the tables or to compute ARLs with headstart values.

It can be shown that the two-sided (V-mask) cusum scheme parameterized by h and k is equivalent to two simultaneously operating one-sided cusum schemes, one that computes an upper cusum and one that computes a lower cusum. Both one-sided schemes use the same parameters h and k.

You can specify h, k, and δ with the options H=, K=, and DELTA= or with the variables _H_, _K_, and _DELTA_ in a LIMITS= data set. The reference value k is optional, and its default value is $k = |\delta|/2$, referred to as the *central reference value*.

Error Probability Approach

This approach is available only for two-sided cusum schemes. Values of α (the probability of incorrectly signaling the occurrence of a shift) and β (the probability of failing to detect a shift) are specified, and h and k are computed from α and β as described in "Defining the V-Mask for a Two-Sided Cusum Scheme" on page 555. The error probability approach interprets the cusum as a sequence of reversed sequential probability ratio tests. Refer to Johnson (1961), Johnson and Leone (1962, 1974), van Dobben de Bruyn (1968), Montgomery (1996), and Wadsworth and others (1986).

Although the error probability method is intuitively appealing, the actual error probabilities achieved may not be close to those specified since the V-mask does not provide for an acceptance region. This has been pointed out by various authors, including Johnson (1961) and van Dobben de Bruyn (1968). If you follow this approach, it is recommended that you examine the average run lengths for the cusum scheme (these are tabulated by the TABLESUMMARY option and are saved in OUTLIMITS= data sets).

You can specify α and β with the ALPHA= and BETA= options or with the variables _ALPHA_ and _BETA_ in a LIMITS= data set. It is not necessary to specify β, and the interpretation of β is somewhat questionable. The SIGMAS= option is an alternative to the ALPHA= option, and the variable _SIGMAS_ is an alternative to the variable _ALPHA_ (if you specify the READSIGMAS option).

Economic Design

The parameters n, h, and k are chosen so that the long-run average cost of the cusum scheme is minimized. Refer to Chiu (1974), Montgomery (1980), Svoboda (1991), and Ho and Case (1994) for reviews of the literature on economic design. This approach typically requires numerical optimization techniques, which are available in SAS/IML software and in the NLP procedure in SAS/OR software.

You can pass the optimal parameters to the CUSUM procedure as values of the variables _LIMITN_, _H_, and _K_ in a LIMITS= data set.

Table 19.18. Average Run Lengths for One-Sided V-Mask Cusum Charts as a Function of h, k, and δ.

Parameters		δ (shift in mean)										
h	k	0.00	0.25	0.50	0.75	1.00	1.50	2.00	2.50	3.00	4.00	5.00
2.50	0.25	27.27	13.43	7.96	5.42	4.06	2.71	2.06	1.68	1.42	1.11	1.01
4.00	0.25	77.08	26.68	13.29	8.38	6.06	3.91	2.93	2.38	2.05	1.61	1.23
6.00	0.25	350.80	51.34	20.90	12.37	8.73	5.51	4.07	3.26	2.74	2.13	1.90
8.00	0.25	736.78	84.00	28.76	16.37	11.39	7.11	5.21	4.15	3.48	2.67	2.14
10.00	0.25	2071.51	124.66	36.71	20.37	14.06	8.71	6.36	5.04	4.20	3.20	2.65
2.00	0.50	38.55	18.19	10.00	6.32	4.45	2.74	1.99	1.58	1.32	1.07	1.01
3.00	0.50	117.60	39.47	17.35	9.68	6.40	3.75	2.68	2.12	1.77	1.31	1.07
4.00	0.50	335.37	77.08	26.68	13.29	8.38	4.75	3.34	2.62	2.19	1.71	1.31
5.00	0.50	930.89	141.69	38.01	17.05	10.38	5.75	4.01	3.11	2.57	2.01	1.69
6.00	0.50	2553.11	250.80	51.34	20.90	12.37	6.75	4.68	3.62	2.98	2.24	1.95
1.50	0.75	42.57	21.09	11.59	7.09	4.78	2.73	1.90	1.48	1.24	1.04	1.00
2.25	0.75	139.71	51.46	22.38	11.66	7.13	3.73	2.51	1.91	1.56	1.16	1.02
3.00	0.75	442.80	117.60	39.47	17.35	9.68	4.73	3.12	2.36	1.93	1.41	1.11
3.75	0.75	1375.71	258.96	65.65	24.16	12.37	5.73	3.71	2.79	2.27	1.72	1.31
4.50	0.75	4251.69	559.95	105.12	32.09	15.15	6.73	4.31	3.21	2.59	1.97	1.60
1.00	1.00	35.29	19.22	11.21	7.03	4.75	2.63	1.78	1.38	1.17	1.02	1.00
1.50	1.00	93.85	42.57	21.09	11.59	7.09	3.50	2.24	1.66	1.34	1.07	1.01
2.00	1.00	258.67	94.34	38.55	18.19	10.00	4.45	2.74	1.99	1.58	1.16	1.02
2.50	1.00	716.00	205.97	68.19	27.27	13.43	5.42	3.25	2.34	1.85	1.31	1.07
3.00	1.00	1962.79	442.80	117.60	39.47	17.35	6.40	3.75	2.68	2.12	1.52	1.16
3.50	1.00	5341.40	943.73	199.57	55.69	21.76	7.39	4.25	3.01	2.37	1.73	1.31
0.70	1.50	67.72	36.03	20.26	12.07	7.63	3.66	2.18	1.55	1.25	1.04	1.00
1.10	1.50	184.28	86.36	42.72	22.50	12.74	5.17	2.80	1.86	1.43	1.08	1.01
1.50	1.50	549.69	221.49	93.85	42.57	21.09	7.09	3.50	2.24	1.66	1.16	1.02
1.90	1.50	1762.09	595.61	210.95	80.54	34.26	9.38	4.26	2.64	1.92	1.29	1.05
2.30	1.50	5897.30	1638.15	476.90	151.04	54.47	12.00	5.03	3.04	2.20	1.45	1.12

Table 19.19. Average Run Lengths for Two-Sided V-Mask Cusum Charts as a Function of h, k, and δ.

Parameters		δ (shift in mean)										
h	k	0.00	0.25	0.50	0.75	1.00	1.50	2.00	2.50	3.00	4.00	5.00
2.50	0.25	13.64	11.22	7.67	5.38	4.06	2.71	2.06	1.68	1.42	1.11	1.01
4.00	0.25	38.54	24.71	13.20	8.38	6.06	3.91	2.93	2.38	2.05	1.61	1.23
6.00	0.25	125.40	50.33	20.89	12.37	8.73	5.51	4.07	3.26	2.74	2.13	1.90
8.00	0.25	368.39	83.63	28.76	16.37	11.39	7.11	5.21	4.15	3.48	2.67	2.14
10.00	0.25	1035.75	124.55	36.71	20.37	14.06	8.71	6.36	5.04	4.20	3.20	2.65
2.00	0.50	19.27	15.25	9.63	6.27	4.44	2.74	1.99	1.58	1.32	1.07	1.01
3.00	0.50	58.80	36.24	17.20	9.67	6.40	3.75	2.68	2.12	1.77	1.31	1.07
4.00	0.50	167.68	74.22	26.63	13.29	8.38	4.75	3.34	2.62	2.19	1.71	1.31
5.00	0.50	465.44	139.49	38.00	17.05	10.38	5.75	4.01	3.11	2.57	2.01	1.69
6.00	0.50	1276.55	249.26	51.34	20.90	12.37	6.75	4.68	3.62	2.98	2.24	1.95
1.50	0.75	21.28	17.22	11.01	7.00	4.77	2.73	1.90	1.48	1.24	1.04	1.00
2.25	0.75	69.85	45.97	22.04	11.63	7.13	3.73	2.51	1.91	1.56	1.16	1.02
3.00	0.75	221.40	110.95	39.31	17.34	9.68	4.73	3.12	2.36	1.93	1.41	1.11
3.75	0.75	687.85	251.56	65.58	24.16	12.37	5.73	3.71	2.79	2.27	1.72	1.31
4.50	0.75	2125.85	552.11	105.09	32.09	15.15	6.73	4.31	3.21	2.59	1.97	1.60
1.00	1.00	17.65	15.03	10.39	6.88	4.72	2.63	1.78	1.38	1.17	1.02	1.00
1.50	1.00	46.92	35.70	20.31	11.49	7.07	3.50	2.24	1.66	1.34	1.07	1.01
2.00	1.00	129.34	84.00	37.93	18.14	10.00	4.45	2.74	1.99	1.58	1.16	1.02
2.50	1.00	358.00	191.48	67.76	27.25	13.43	5.42	3.25	2.34	1.85	1.31	1.07
3.00	1.00	981.39	423.29	117.32	39.47	17.35	6.40	3.75	2.68	2.12	1.52	1.16
3.50	1.00	2670.70	917.89	199.40	55.69	21.76	7.39	4.25	3.01	2.37	1.73	1.31
0.70	1.50	33.86	28.41	18.90	11.84	7.59	3.66	2.18	1.55	1.25	1.04	1.00
1.10	1.50	92.14	71.41	40.91	22.29	12.71	5.17	2.80	1.86	1.43	1.08	1.01
1.50	1.50	274.84	191.58	91.58	42.39	21.07	7.09	3.50	2.24	1.66	1.16	1.02
1.90	1.50	881.05	536.07	208.31	80.41	34.25	9.38	4.26	2.64	1.92	1.29	1.05
2.30	1.50	2948.65	1523.15	474.09	150.96	54.47	12.00	5.03	3.04	2.20	1.45	1.12

Cusum Charts Compared with Shewhart Charts

Although cusum charts and Shewhart charts are both used to detect shifts in the process mean, there are important differences in the two methods.

- Each point on a Shewhart chart is based on information for a single subgroup sample or measurement. Each point on a cusum chart is based on information from all samples (measurements) up to and including the current sample (measurement).

- On a Shewhart chart, upper and lower control limits are used to decide whether a point signals an out-of-control condition. On a cusum chart, the limits take the form of a decision interval or a V-mask.
- On a Shewhart chart, the control limits are commonly computed as 3σ limits. On a cusum chart, the limits are determined from average run length specifications, specified error probabilities, or an economic design.

A cusum chart offers several advantages over a Shewhart chart.

- A cusum chart is more efficient for detecting small shifts in the process mean, in particular, shifts of 0.5 to 2 standard deviations from the target mean (refer to Montgomery 1996). Lucas (1976) noted that "a V-mask designed to detect a 1σ shift will detect it about four times as fast as a competing Shewhart chart."
- Shifts in the process mean are visually easy to detect on a cusum chart since they produce a change in the slope of the plotted points. The point at which the slope changes is the point at which the shift has occurred.

These advantages are not as pronounced if the Shewhart chart is augmented by the tests for special causes described by Nelson (1984, 1985). Also see Chapter 55, "Tests for Special Causes." Moreover,

- cusum schemes are more complicated to design.
- a cusum chart can be slower to detect large shifts in the process mean.
- it can be difficult to interpret point patterns on a cusum chart since the cusums are correlated.

Methods for Estimating the Standard Deviation

It is recommended practice to provide a stable estimate or standard value for σ with either the SIGMA0= option or the variable _STDDEV_ in a LIMITS= data set. However, if such a value is not available, you can compute an estimate $\hat{\sigma}$ from the data, as described in this section.

This section provides formulas for various methods used to estimate the standard deviation σ. One method is applicable with individual measurements, and three are applicable with subgrouped data. The methods can be requested with the SMETHOD= option.

Method for Individual Measurements

When the cumulative sums are calculated from individual observations

$$x_1, x_2, \ldots, x_N$$

rather than subgroup samples of two or more observations, the CUSUM procedure estimates σ as $\sqrt{\hat{\sigma}^2}$, where

$$\hat{\sigma}^2 = \frac{1}{2(N-1)} \sum_{i=1}^{N-1} (x_{i+1} - x_i)^2$$

where N is the number of observations. Wetherill (1977) states that the estimate of the variance is biased if the measurements are autocorrelated.

Note that you can compute alternative estimates (for instance, robust estimates or estimates based on variance components models) by analyzing the data with SAS modeling procedures or your own DATA step program. Such estimates can be passed to the CUSUM procedure as values of the variable _STDDEV_ in a LIMITS= data set.

NOWEIGHT Method for Subgroup Samples

This method is the default for cusum charts for subgrouped data. The estimate is

$$\hat{\sigma} = \frac{(s_1/c_4(n_1)) + \cdots + (s_N/c_4(n_N))}{N}$$

where n_i is the sample size of the i^{th} subgroup, N is the number of subgroups for which $n_i \geq 2$, s_i is the sample standard deviation of the observations $x_{i1}, \ldots, x_{in_i}$ in the i^{th} subgroup.

$$s_i = \sqrt{(1/(n_i-1)) \sum_{j=1}^{n_i} (x_{ij} - \bar{X}_i)^2}$$

and

$$c_4(n_i) = \frac{\Gamma(n_i/2)\sqrt{2/(n_i-1)}}{\Gamma((n_i-1)/2)}$$

where $\Gamma(\cdot)$ denotes the gamma function, and $\bar{X}_i$ denotes the i^{th} subgroup mean. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$. If the observations are normally distributed, then the expected value of s_i is

$$\mathrm{E}(s_i) = c_4(n_i)\sigma$$

Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the *ASTM Manual on Presentation of Data and Control Chart Analysis*.

MVLUE Method for Subgroup Samples

If you specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed, as introduced by Burr (1969, 1976). This estimate is a weighted average of unbiased estimates of σ of the form

$$s_i/c_4(n_i)$$

where

s_i	is the standard deviation of the i^{th} subgroup.
$c_4(n_i)$	is the unbiasing factor defined previously.
n_i	is the i^{th} subgroup sample size, $i = 1, 2, \ldots, N$.
N	is the number of subgroups for which $n_i \geq 2$.

The estimate is

$$\hat{\sigma} = \frac{h_1 s_1/c_4(n_1) + \cdots + h_N s_N/c_4(n_N)}{h_1 + \cdots + h_N}$$

where $h_i = c_4^2(n_i)/(1 - c_4^2(n_i))$. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$.

The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes and is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate (NOWEIGHT).

RMSDF Method for Subgroup Samples

If you specify SMETHOD=RMSDF, a weighted root-mean-square estimate is computed:

$$\hat{\sigma} = \frac{\sqrt{(n_1 - 1)s_1^2 + \cdots + (n_N - 1)s_N^2}}{c_4(n)\sqrt{n_1 + \cdots + n_N - N}}$$

where

n_i	is the sample size of the i^{th} subgroup.
N	is the number of subgroups for which $n_i \geq 2$.
s_i	is the sample standard deviation of the i^{th} subgroup.
$c_4(n_i)$	is the unbiasing factor defined previously.
n	is equal to $(n_1 + \cdots + n_N) - (N - 1)$.

The weights in the root-mean-square expression are the degrees of freedom $n_i - 1$. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$.

If the unknown standard deviation σ is constant across subgroups, the root-mean-square estimate is more efficient than the minimum variance linear unbiased estimate. However, as noted by Burr (1969), "the constancy of σ is the very thing under test," and if σ varies across subgroups, the root-mean-square estimate tends to be more inflated than the MVLUE.

Output Data Sets

OUTLIMITS= Data Set

When you save the parameters for the cusum scheme in an OUTLIMITS= data set, the following variables are included:

Table 19.20. OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of Type 1 error
ARLIN	average run length for zero shift
ARLOUT	average run length for shift of δ
BETA	probability (β) of Type 2 error
DELTA	shift (δ) to be detected
_H_	decision interval h for one-sided scheme; distance h between origin and upper arm V-mask for two-sided scheme
HSTART	headstart value
INDEX	optional identifier for cusum parameters (if the OUTINDEX= option is specified)
_K_	reference value k for one-sided scheme; slope of lower V-mask arm for two-sided scheme
LIMITN	nominal sample size for cusum scheme
MEAN	estimated process mean ($\overline{\overline{X}}$)
_MU0_	target mean μ_0
ORIGIN	origin of V-mask
SCHEME	type of scheme (`ONESIDED` or `TWOSIDED`)
SIGMAS	$z_{1-\alpha/2}$
STDDEV	estimated or known standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*subgroup-variable* specified in XCHART statement
TYPE	type (`ESTIMATE` or `STANDARD`) of _STDDEV_
VAR	*process* specified in XCHART statement

Notes:

1. If the subgroup sample sizes vary, the special missing value *V* is assigned to the variable _LIMITN_.
2. If a V-mask is specified with SIGMAS=k, _ALPHA_ is computed as $\alpha = 2(1 - \Phi(k))$, where $\Phi(\cdot)$ is the standard normal distribution function.
3. If a V-mask is specified with ALPHA=α, _SIGMAS_ is computed as $k = \Phi^{-1}(1 - \alpha/2)$, where Φ^{-1} is the inverse standard normal distribution function.
4. BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the XCHART statement. For an example, see "Saving Cusum Scheme Parameters" on page 531.

OUTHISTORY= Data Set

When you save subgroup summary statistics in an OUTHISTORY= data set, the following variables are included:

- the *subgroup-variable*
- a subgroup mean variable named by *process* suffixed with *X*
- a subgroup sample size variable named by *process* suffixed with *N*
- a subgroup standard deviation variable named by *process* suffixed with *S*
- a cusum variable named by *process* suffixed with *C*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Variables containing subgroup summary statistics are created for each *process* specified in the XCHART statement. For example, consider the following statements:

```
proc cusum data=steel limits=stparm;
   xchart (width diameter)*lot / outhistory=summary;
run;
```

The data set SUMMARY would contain nine variables named LOT, WIDTHX, WIDTHS, WIDTHN, WIDTHC, DIAMTERX, DIAMTERS, DIAMTERN, and DIAMTERC.

Additionally, if specified, the following variables are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example creating an OUTHISTORY= data set, see "Saving Summary Statistics" on page 527.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup means, subgroup sample sizes, cusums, and cusum limits. The following variables are included:

Variable	Description
CUSUM	cumulative sum
EXLIM	decision interval or V-mask arm exceeded
_H_	decision interval
MASKL	lower arm of V-mask
MASKU	upper arm of V-mask
subgroup	values of the subgroup variable
SUBN	subgroup sample size
SUBX	subgroup mean
SUBS	subgroup standard deviation
VAR	*process* specified in XCHART statement

In addition, the following variables are saved if specified:

- BY variables
- *block-variables*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)
- _TREND_ (if the TRENDVAR= option is specified)
- *symbol-variable*

Note that the variables _VAR_ and _EXLIM_ are character variables of length eight. The variable _PHASE_ is a character variable of length 16.

ODS Tables

The following table summarizes the ODS tables that you can request with the XCHART statement.

Table 19.21. ODS Tables Produced with the XCHART Statement

Table Name	Description	Options
CompCusum	computational form of the cusum scheme	TABLEALL, TABLECOMP
Parameters	cusum parameters and computed average run lengths	TABLEALL, TABLESUMMARY
XCHART	cusum chart summary statistics	TABLEALL, TABLECHART, TABLEOUT

Input Data Sets

DATA= Data Set

You can read raw data (measurements) from a DATA= data set specified in the PROC CUSUM statement. Each *process* specified in the XCHART statement must be a SAS variable in the DATA= data set. The values of this variable are typically measurements of a quality characteristic taken on items in subgroup samples indexed by the values of the subgroup variable. The *subgroup-variable* specified in the XCHART

statement must also be a SAS variable in the DATA= data set. Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

Each observation in a DATA= data set should contain a raw measurement for each *process* and a value for the subgroup variable. If the i^{th} subgroup contains n_i items, there should be n_i consecutive observations for which the value of the subgroup variable is the index of the i^{th} subgroup. For example, if each of 30 subgroup samples contains five items, the DATA= data set should contain 150 observations.

By default, the CUSUM procedure reads all of the observations in a DATA= data set. However, if the DATA= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option in the XCHART statement.

For an example of a DATA= data set, see "Creating a V-Mask Cusum Chart from Raw Data" on page 522.

LIMITS= Data Set

You can read cusum scheme parameters from a LIMITS= data set specified in the PROC CUSUM statement.* As an alternative to specifying the parameters with options, a LIMITS= data set provides the following advantages: it facilitates reusing a permanently saved set of parameters, reading a distinct set of parameters for each *process* specified in the XCHART statement, and keeping track of multiple sets of parameters for the same *process* over time.

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the CUSUM procedure. Such data sets always contain the variables required for a LIMITS= data set; consequently, this is the easiest way to construct a LIMITS= data set.

A LIMITS= data set can also be created directly using a DATA step. The variables required for the data set depend on the type of cusum scheme and how the scheme is specified. The following restrictions apply:

- The variables _VAR_, _SUBGRP_, _DELTA_, and _MU0_ are required.
- For a one-sided cusum scheme, _H_ is required.
- For a two-sided cusum scheme, one of the following three variables is required: _ALPHA_, _H_, or _SIGMAS_.

*If you are using Release 6.09 or an earlier release, you must also specify the READLIMITS or READINDEX= option in the XCHART statement.

- If you plan to use the READINDEX= option, the variable _INDEX_ is required; otherwise, it is optional.
- For a one-sided scheme, the variable _SCHEME_ is required; otherwise, it is optional.
- If you want to provide a value for the process standard deviation σ, the variable _STDDEV_ is required; otherwise, it is optional.

Variable names in a LIMITS= data set are predefined; the procedure reads only variables with these predefined names. With the exception of BY variables, all names start and end with an underscore. In addition, note the following:

- The variables _VAR_, _SUBGRP_, _TYPE_, and _SCHEME_ must be character variables of length eight. The variable _INDEX_ must be a character variable of length 16.
- The variable _TYPE_ is a bookkeeping variable that uses the values `ESTIMATE` and `STANDARD` to record whether the value of _STDDEV_ represents an estimate or standard (known) value.
- BY variables are required if specified with a BY statement.

For an example of reading control limit information from a LIMITS= data set, see "Reading Cusum Scheme Parameters" on page 533.

HISTORY= Data Set

Instead of reading raw data from a DATA= data set, you can read subgroup summary statistics from a HISTORY= data set specified in the PROC CUSUM statement. This allows you to reuse OUTHISTORY= data sets that have been created in previous runs of the CUSUM, MACONTROL, or SHEWHART procedures or to read output data sets created with SAS summarization procedures such as PROC MEANS. A HISTORY= data set must contain the following variables:

- *subgroup-variable*
- subgroup mean variable for each *process*
- subgroup standard deviation variable for each *process*
- subgroup sample size variable for each *process*

The names of the subgroup mean, subgroup standard deviation, and subgroup sample size variables must be the *process* concatenated with the special suffix characters X, S, and N respectively.

For example, consider the following statements:

```
proc cusum history=steel limits=stlparm;
   xchart (weight yldstren)*batch;
run;
```

The data set STEEL must contain the variables BATCH, WEIGHTX, WEIGHTS, WEIGHTN, YLDSRENX, YLDSRENS, and YLDSRENN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the CUSUM procedure reads all of the observations in a HISTORY= data set. However, if the HISTORY= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as phases) by specifying the READPHASES= option.

For an example of reading summary information from a HISTORY= data set, see "Creating a V-Mask Cusum Chart from Subgroup Summary Data" on page 525.

Missing Values

An observation read from a DATA= or HISTORY= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples

This section provides advanced examples of the XCHART statement.

Example 19.1. Cusum and Standard Deviation Charts

See CUSXS in the SAS/QC Sample Library

When you are working with subgrouped data, it can be helpful to accompany a cusum chart for means with a Shewhart s chart for monitoring the variability of the process. This example creates this combination for the variable WEIGHT in the data set OIL (see "Creating a V-Mask Cusum Chart from Raw Data" on page 522).

The first step is to create a one-sided cusum chart for means that detects a shift of one standard error ($\delta = -1$) below the target mean.

```
proc cusum data=oil;
   xchart weight*hour /
      nochart
      mu0=8.100         /* target mean for process  */
      sigma0=0.050      /* known standard deviation */
      delta=-1          /* shift to be detected     */
      h=3               /* cusum parameter h        */
      k=0.5             /* cusum parameter k        */
      scheme=onesided
      outtable = tabcusum
         ( drop   = _var_ _subn_ _subx_ _exlim_
           rename = ( _cusum_ = _subx_ _h_ = _uclx_ ) )
      ;
run;
```

The results are saved in an OUTTABLE= data set named TABCUSUM. The cusum variable (_CUSUM_) and the decision interval variable (_H_) are renamed to _SUBX_ and _LCLX_ so that they can later be read by the SHEWHART procedure.

The next step is to construct a Shewhart $\bar{X}$ and s chart for WEIGHT and save the results in a data set named TABXSCHT.

```
proc shewhart data=oil;
   xschart weight*hour /
      nochart
      outtable = tabxscht
         ( drop = _subx_  _uclx_ );
run;
```

Note that the variables _SUBX_ and _UCLX_ are dropped from TABXSCHT.

The third step is to merge the data sets TABCUSUM and TABXSCHT.

```
data taball;
   merge tabxscht tabcusum; by hour;
   _mean_ = _uclx_ * 0.5;
   _lclx_ = 0.0;
run;

title ;
proc print;
run;
```

The variable _LCLX_ is assigned the role of the lower limit for the cusums, and the variable _MEAN_ is assigned a dummy value. Now, TABALL, which is listed in Output 19.1.1, has the structure required for a TABLE= data set used with the XSCHART statement in the SHEWHART procedure (see "TABLE= Data Set" on page 1821 in Chapter 51, "XSCHART Statement,").

Output 19.1.1. Listing of the Data Set TABALL

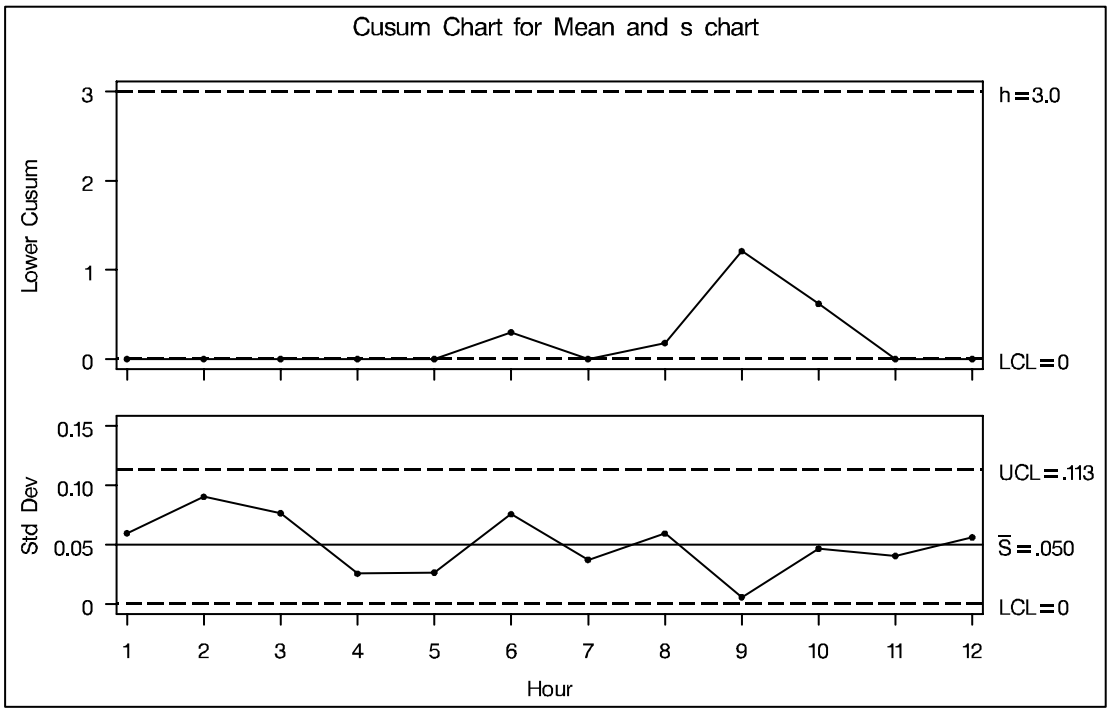
Obs	_VAR_	hour	_SIGMAS_	_LIMITN_	_SUBN_	_LCLX_	_MEAN_	_EXLIM_	_LCLS_	_SUBS_	_S_	_UCLS_	_EXLIMS_	_subx_	_uclx_
1	weight	1	3	4	4	0	1.5		0	0.059640	0.049943	0.11317		0.00	3
2	weight	2	3	4	4	0	1.5		0	0.090220	0.049943	0.11317		0.00	3
3	weight	3	3	4	4	0	1.5		0	0.076346	0.049943	0.11317		0.00	3
4	weight	4	3	4	4	0	1.5		0	0.025552	0.049943	0.11317		0.00	3
5	weight	5	3	4	4	0	1.5		0	0.026500	0.049943	0.11317		0.00	3
6	weight	6	3	4	4	0	1.5		0	0.075617	0.049943	0.11317		0.30	3
7	weight	7	3	4	4	0	1.5		0	0.037242	0.049943	0.11317		0.00	3
8	weight	8	3	4	4	0	1.5		0	0.059290	0.049943	0.11317		0.18	3
9	weight	9	3	4	4	0	1.5		0	0.005737	0.049943	0.11317		1.21	3
10	weight	10	3	4	4	0	1.5		0	0.046522	0.049943	0.11317		0.62	3
11	weight	11	3	4	4	0	1.5		0	0.040542	0.049943	0.11317		0.00	3
12	weight	12	3	4	4	0	1.5		0	0.056103	0.049943	0.11317		0.00	3

The final step is to use the SHEWHART procedure to read TABALL as a TABLE= data set and to display the cusum and *s* charts.

```
title 'Cusum Chart for Mean and s chart';
proc shewhart table=taball;
   xschart weight * hour /
      nolimitslegend
      ucllabel = 'h=3.0'
      noctl
      split = '/'
      nolegend ;
   label _subx_ = 'Lower Cusum/Std Dev';
run;
```

The central line for the primary (cusum) chart is suppressed with the NOCTL option, and the default *3σ Limits* legend is suppressed with the NOLIMITLEGEND option. The charts are shown in Output 19.1.2.

Output 19.1.2. Combined Cusum Chart and s Chart



The process variability is stable, and there is no signal of a downward shift in the process mean.

Example 19.2. Upper and Lower One-Sided Cusum Charts

See CUSUPLO in the SAS/QC Sample Library

This example illustrates how to combine upper and lower one-sided cusum charts for means in the same display. As in the preceding example, OUTTABLE= data sets are created with the CUSUM procedure, and the display is created with the SHEWHART procedure.

The following statements analyze the variable WEIGHT in the data set OIL (see "Creating a V-Mask Cusum Chart from Raw Data" on page 522). The first step is to compute and save upper and lower one-sided cusums for shifts of one standard error in the positive and negative directions.

```
proc cusum data=oil;
   xchart weight*hour /
      nochart
      mu0=8.100          /* target mean for process  */
      sigma0=0.050       /* known standard deviation */
      delta=1            /* shift to be detected     */
      h=3                /* cusum parameter h        */
      k=0.5              /* cusum parameter k        */
      scheme=onesided
      outtable = tabupper
         ( drop   = _subx_ _subs_ _exlim_
           rename = ( _cusum_ = _subx_ _h_ = _uclx_ ) )
      ;
```

```
   xchart weight*hour /
      nochart
      mu0=8.100         /* target mean for process  */
      sigma0=0.050      /* known standard deviation */
      delta=-1          /* shift to be detected     */
      h=3               /* cusum parameter h        */
      k=0.5             /* cusum parameter k        */
      scheme=onesided
      outtable = tablower
         ( drop   = _var_ _subn_ _limitn_ _subx_ _subs_ _exlim_
           rename = ( _cusum_ = _subs_ _h_ = _ucls_ ) )
      ;
run;
```

Next, the OUTTABLE= data sets are merged.

```
data tabboth;
   merge tabupper tablower; by hour;
   _mean_    = _uclx_ * 0.5;
   _s_       = _ucls_ * 0.5;
   _lclx_    = 0.0;
   _lcls_    = 0.0;
run;
```

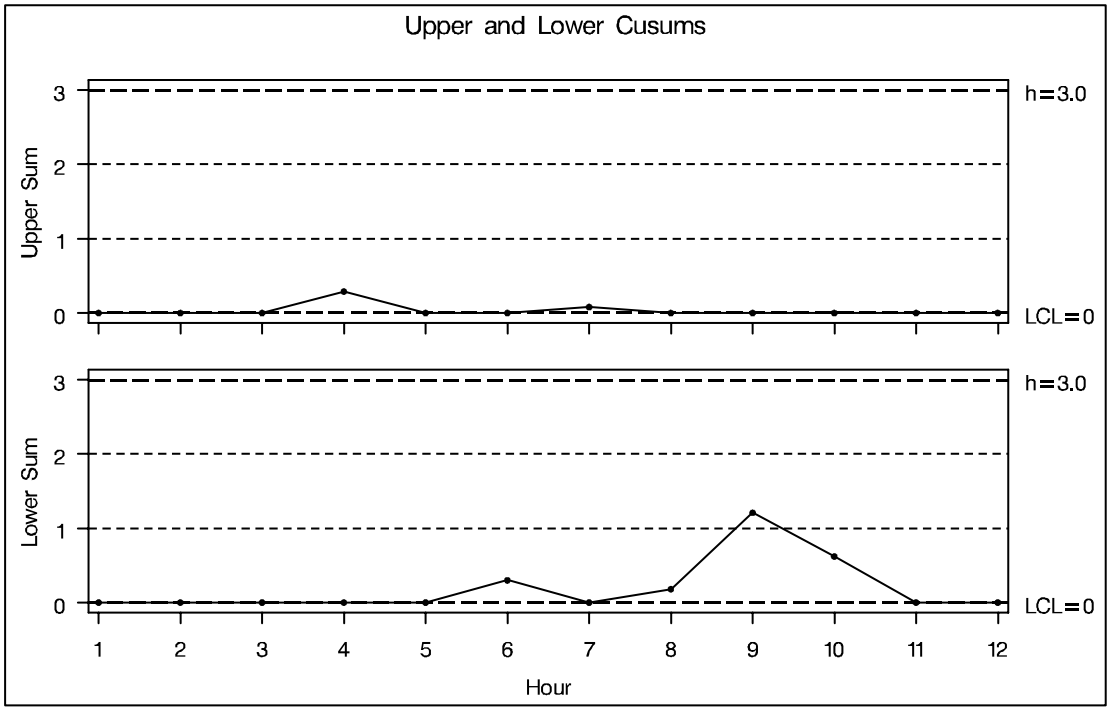
The variables _LCLX_ and _UCLX_ are assigned lower limits of zero for the cusums, and the variables _MEAN_ and _S_ are assigned dummy values. Now, TABBOTH has the structure required for a TABLE= data set used with the XSCHART statement in the SHEWHART procedure (see “TABLE= Data Set” on page 1821 in Chapter 51, “XSCHART Statement,”).

The final step is to read TABBOTH as a TABLE= data set with the SHEWHART procedure.

```
title 'Upper and Lower Cusums';
proc shewhart table=tabboth;
   xschart weight * hour /
      nolimitslegend
      ucllabel  = 'h=3.0'
      ucllabel2 = 'h=3.0'
      ypct1     = 50
      vref      = 1 2
      vref2     = 1 2
      lvref     = 2
      noctl
      noctl2
      split = '/'
      nolegend ;
   label _subx_ = 'Upper Sum/Lower Sum';
run;
```

The combined display is shown in Output 19.2.1. There is no evidence of a shift in either direction.

Output 19.2.1. Upper and Lower One-Sided Cusum Charts



Example 19.3. Combined Shewhart–Cusum Scheme

See CUSCOMB in the SAS/QC Sample Library

Lucas (1982) introduced a combined Shewhart-cusum scheme that is illustrated in this example. Also refer to Ryan (1989). The data set used here is CANS, which is created in "Creating a One-Sided Cusum Chart with a Decision Interval" on page 528.

The first step is to compute and save one-sided cusums to detect a positive shift from the mean.

```
proc cusum data=cans;
   xchart weight*hour /
      nochart
      mu0    = 8.100     /* target mean for process  */
      sigma0 = 0.050     /* known standard deviation */
      delta  = 1         /* shift to be detected     */
      h      = 3         /* cusum parameter h        */
      k      = 0.5       /* cusum parameter k        */
      scheme = onesided
      outtable = tabcus
         ( drop   = _var_ _subn_ _exlim_
           rename = ( _cusum_ = _subr_ _h_ = _uclr_ ) )
      ;
run;
```

Note that a headstart value is not used here but can be specified with the HSTART= option. Several variables in the OUTTABLE= data set are dropped or renamed so that they can later be read by the SHEWHART procedure.

The next step is to construct a Shewhart chart (not shown) for individual measurements.

```
proc shewhart data=cans;
   irchart weight*hour /
      nochart
      mu0      = 8.100
      sigma0   = 0.050
      outtable = tabx
         ( drop   = _subr_  _lclr_ _r_ _uclr_ );
   id comment;
run;
```

By default, 3σ limits are computed, but the multiple of σ can be modified with the SIGMAS= option. As before, the results are saved in an OUTTABLE= data set.

Next, the two OUTTABLE= data sets are merged.

```
data combine;
   merge tabx tabcus; by hour;
   _lclr_ = 0.0;
   _r_    = 0.5 * _uclr_;
run;
```

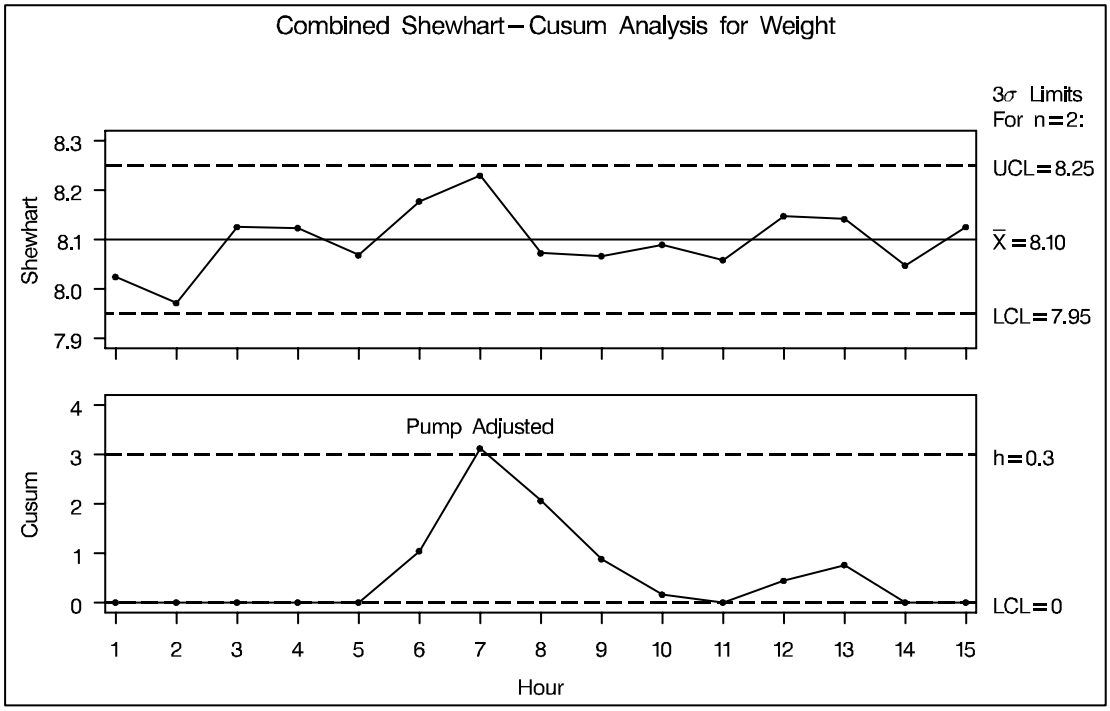
The data set COMBINE has the structure required for a TABLE= data set used with the IRCHART statement in the SHEWHART procedure (see "TABLE= Data Set" on page 1378 in Chapter 41, "IRCHART Statement,").

Finally, the combined scheme is displayed with the SHEWHART procedure.

```
title "Combined Shewhart-Cusum Analysis for Weight";
proc shewhart table=combine;
   irchart weight*hour /
      ypct1     = 50
      noctl2
      ucllabel2 = 'h=0.3'
      outlabel1 = ( comment )
      outlabel2 = ( comment )
      split     = '/';
   label _subi_ = 'Shewhart/Cusum';
run;
```

The chart is shown in Output 19.3.1.

Output 19.3.1. Combined Shewhart–Cusum Scheme



Note that a shift is detected by the cusum scheme but not by the Shewhart chart. The point exceeding the decision interval is labeled with the variable COMMENT created in the data set CANS.

Lucas (1982) tabulates average run lengths for combined Shewhart-cusum schemes. The scheme used here has an ARL of 111.1 for $\delta = 0$ and an ARL of 6.322 for $\delta = 1$.

Chapter 20
INSET Statement

Chapter Contents

OVERVIEW 579

GETTING STARTED 579

SYNTAX 581

Chapter 20
INSET Statement

Overview

The INSET statement allows you to enhance a cusum chart by adding a box or table (referred to as an *inset*) of summary statistics directly to the graph. A possible application of an inset is to present cusum parameters on the chart rather than displaying them in a legend. An inset can also display arbitrary values provided in a SAS data set.

Note that the INSET statement by itself does not produce a display but must be used in conjunction with an XCHART statement. Insets are not available with line printer output, so the INSET statement is not applicable when the LINEPRINTER option is specified in the PROC CUSUM statement.

You can use options in the INSET statement to

- specify the position of the inset
- specify a header for the inset table
- specify graphical enhancements, such as background colors, text colors, text height, text font, and drop shadows

Getting Started

This section introduces the INSET statement with a basic example showing how it is used. See Chapter 52, "INSET and INSET2 Statements," for a complete description of the INSET statement.

This example is based on the same scenario as the first example in the "Getting Started" section of Chapter 19, "XCHART Statement." A machine fills cans with oil additive and a two-sided cusum chart is used to detect shifts from the target mean of 8.100 ounces. The following statements create the data set OIL and request a two-sided cusum chart with an inset:

```
data oil;
   label hour = 'Hour';
   input hour @;
   do i=1 to 4;
      input weight @;
      output;
      end;
   drop i;
   datalines;
 1  8.024  8.135  8.151  8.065
```

```
 2   7.971   8.165   8.077   8.157
 3   8.125   8.031   8.198   8.050
 4   8.123   8.107   8.154   8.095
 5   8.068   8.093   8.116   8.128
 6   8.177   8.011   8.102   8.030
 7   8.129   8.060   8.125   8.144
 8   8.072   8.010   8.097   8.153
 9   8.066   8.067   8.055   8.059
10   8.089   8.064   8.170   8.086
11   8.058   8.098   8.114   8.156
12   8.147   8.116   8.116   8.018
;


symbol v=dot;
title 'Cusum Chart for Average Weights of Cans';
proc cusum data=oil;
   xchart weight*hour /
      mu0     = 8.100           /* Target mean for process  */
      sigma0  = 0.050           /* Known standard deviation */
      delta   = 1               /* Shift to be detected     */
      alpha   = 0.10            /* Type I error probability */
      vaxis   = -5 to 3
      nolegend;
   label weight = 'Cumulative Sum';
   inset arl0 alpha delta h k mu0 shift sigmas / pos = sw;
run;
```

The resulting cusum chart is shown in Figure 20.1.

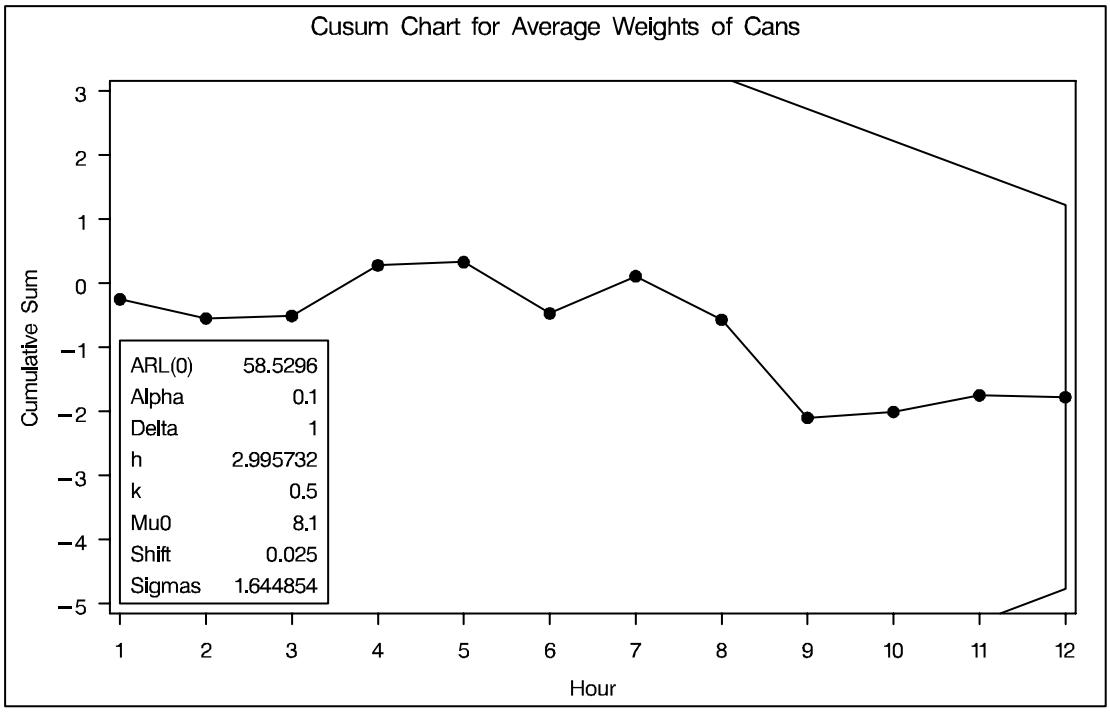


Figure 20.1. Two-Sided Cusum Chart with an Inset

Syntax

The syntax for the INSET statement is as follows:

INSET *keyword-list* < **/** *options* >**;**

You can use any number of INSET statements in the CUSUM procedure. Each INSET statement produces a separate inset and must follow an XCHART statement. The inset appears on every panel (page) produced by the last XCHART statement preceding it.

Keywords specify the statistics to be displayed in an inset; options control the inset's location and appearance. A complete description of the INSET statement syntax is given starting on page 1841 of Chapter 52 The INSET statement options are identical in the CUSUM and SHEWHART procedures, but the available keywords are different. The keywords available with the CUSUM procedure are listed in Table 20.1 to Table 20.3.

Table 20.1. Summary Statistics

ARL0	average run length for zero shift
ARLDELTA	average run length for shift of δ
DATA=	arbitrary values from *SAS-data-set*
N	nominal subgroup size
NMIN	minimum subgroup size
NMAX	maximum subgroup size

Table 20.2. Parameters for One-Sided (Decision Interval) Cusum Scheme

DELTA	shift to be detected as multiple of standard error
H	decision interval h as a multiple of standard error
HEADSTART	headstart value S_0 as a multiple of standard error
K	reference value k
MU0	target mean μ_0
SHIFT	shift to be detected in data units
STDDEV	estimated or specified process standard deviation

Table 20.3. Parameters for Two-Sided (V-Mask) Cusum Scheme

ALPHA	probability of Type 1 error
BETA	probability of Type 2 error
H	vertical distance between V-mask origin and upper (or lower) arm
K	slope of lower arm of V-mask
SIGMAS	probability of Type 1 error as probability that standard normally distributed variable exceeds a specified value in absolute value

References

American Society for Quality Control (1983), *ASQC Glossary and Tables for Statistical Quality Control*, 230 W. Wells Street, Milwaukee, Wisconsin 53203.

American Society for Testing and Materials (1976), *ASTM Manual on Presentation of Data and Control Chart Analysis*, 1916 Race Street, Philadelphia, PA 19103.

Burr, I. W. (1969), "Control Charts for Measurements with Varying Sample Sizes," *Journal of Quality Technology*, 1, 163–167.

Burr, I. W. (1976), *Statistical Quality Control Methods, Volume 16*, New York: Marcel Dekker, Inc.

Burr, I. W. (1979), *Elementary Statistical Quality Control, Volume 25*, New York: Marcel Dekker, Inc.

Chiu, W. K. (1974), "The Economic Design of Cusum Charts for Controlling Normal Means," *Applied Statistics*, 23, 420–433.

Duncan, A. J. (1974), *Quality Control and Industrial Statistics, Fourth Edition*, Homewood, Illinois: Richard D. Irwin, Inc.

Goel, A. L. (1982), "Cumulative Sum Control Charts," *Encyclopedia of Statistical Sciences, Volume 2*. New York: John Wiley & Sons, Inc.

Goel, A. L. and Wu, S. M. (1971), "Determination of A.R.L. and a Contour Nomogram for Cusum Charts to Control Normal Mean," *Technometrics*, 13, 221–230.

Ho, C. and Case, K. E. (1994), "Economic Design of Control Charts: A Literature Review for 1981–1991," *Journal of Quality Technology*, 26, 39–53.

Johnson, N. L. (1961), "A Simple Theoretical Approach to Cumulative Sum Control Chart," *Journal of the American Statistical Association*, 56, 835–840.

Johnson, N. L. and Leone, F. C. (1962), "Cumulative Sum Control Charts: Mathematical Principles Applied to Their Construction and Use," *Industrial Quality Control*, 18, June, 15–21; July, 29–36; August, 22–28.

Johnson, N. L. and Leone, F. C. (1974), *Statistics and Experimental Design, Second Edition, Volume 1*, New York: John Wiley & Sons, Inc.

Kemp, K. W. (1961), "The Average Run Length of the Cumulative Sum Control Chart When a 'V' Mask Is Used," *Journal of the Royal Statistical Society, Series B*, 23, 149–153.

Kume, H. (1985), *Statistical Methods for Quality Improvement*, Tokyo: AOTS Chosakai, Ltd.

Lucas, J. M. (1976), "The Design and Use of V-Mask Control Schemes," *Journal of Quality Technology*, 8, 1–12.

Lucas, J. M. (1982), "Combined Shewhart–CUSUM Quality Control Schemes," *Journal of Quality Technology*, 14, 51–59.

Lucas, J. M. and Crosier, R. B. (1982), "Fast Initial Response for CUSUM Quality Control Schemes: Give Your CUSUM a Head Start," *Technometrics*, 24, 199–205.

Montgomery, D. C. (1980), "The Economic Design of Control Charts: A Review and Literature," *Journal of Quality Technology*, 12, 75–87.

Montgomery, D. C. (1996), *Introduction to Statistical Quality Control, Third Edition*, New York: John Wiley & Sons, Inc.

Nelson, L. (1984), "The Shewhart Control Chart—Tests for Special Causes," *Journal of Quality Technology*, 15, 237–239.

Nelson, L. (1985), "Interpreting Shewhart $\bar{X}$ Control Charts," *Journal of Quality Technology*, 17, 114–116.

Ryan, T. (1989), *Statistical Methods for Quality Improvement*, New York: John Wiley & Sons, Inc.

SAS Institute, Inc. (1999), *SAS/GRAPH Software: Reference, Version 8*, Cary, NC: SAS Institute Inc.

Svoboda, L. (1991), "Economic Design of Control Charts: A Review and Literature Survey (1979–1989)," in *Statistical Process Control in Manufacturing*, edited by J. B. Keats and D. C. Montgomery, New York: Marcel Dekker.

van Dobben de Bruyn, C. S. (1968), *Cumulative Sum Tests: Theory and Practice, Griffin's Statistical Monographs & Courses, No. 24*, New York: Hafner.

Wadsworth, H. M., Stephens, K. S., and Godfrey, A. B. (1986), *Modern Methods for Quality Control and Improvement*, New York: John Wiley & Sons, Inc.

Wetherill, G. B. (1977), *Sampling Inspection and Quality Control, Second Edition*, New York: Chapman and Hall.

Wetherill, G. B. and Brown, D. B. (1991), *Statistical Process Control: Theory and Practice*, London: Chapman and Hall.

Part 4
The FACTEX Procedure

Contents

Chapter 21. Introduction 587

Chapter 22. Details of the FACTEX Procedure 599

Chapter 23. Theory of Orthogonal Designs 661

References 669

Chapter 21
Introduction

Chapter Contents

OVERVIEW 589
Features 589
Learning about the FACTEX Procedure 590

GETTING STARTED 590
Example of a Two-Level Full Factorial Design 591
Example of a Full Factorial Design in Two Blocks 593
Example of a Half-Fraction Factorial Design 595
Using the FACTEX Procedure Interactively 597

Chapter 21
Introduction to the FACTEX Procedure

Overview

The FACTEX procedure constructs orthogonal factorial experimental designs. These designs can be either full or fractional factorial designs, and they can be with or without blocks. Once you have constructed a design with the FACTEX procedure and run the experiment, you can analyze the results with a variety of SAS procedures including PROC GLM and PROC REG.

Factorial experiments are useful for studying the effects of various factors on a response. Texts that discuss experimental design include Box, Hunter, and Hunter (1978); Cochran and Cox (1957); Hogg and Ledolter (1992); Mason, Gunst, and Hess (1989); and Montgomery (1996). For details on the general mathematical theory of orthogonal factorial designs, refer to Bose (1947).

Note: For two-level designs, instead of using FACTEX directly, a more appropriate tool for you may be the ADX Interface. The ADX Interface, which has been completely revised in Version 7, is designed primarily for engineers and researchers who require a point-and-click solution for the entire experimental process, from building the designs through determining significant effects to optimization and reporting. ADX gives you most of the two-level designs provided by the FACTEX procedure in a system that integrates construction and analysis of designs and without the need for programming. In addition to two-level designs for standard models (with and without blocking), ADX makes it easy to use FACTEX to construct designs for estimating particular effects of interest. Moreover, ADX also uses the OPTEX procedure to construct two-level designs of non-standard sizes. For more information, see Chapter 1, "Overview of ADX," (*Getting Started with the SAS 9 ADX Interface for Design of Experiments*) .

Features

There is no inherent limit to the number of factors and the size of the design that you can construct using the FACTEX procedure. Instead of looking up designs in an internal table, the FACTEX procedure uses a general algorithm to search for the construction rules for a specified design.

You can use the FACTEX procedure to generate designs such as the following:

- factorial designs, such as 2^3 designs, with and without blocking
- fractional factorial designs, such as 2^{4-1}_{IV}, with and without blocking
- three-level designs, with and without blocking
- mixed-level factorial designs, such as 4×3 designs, with and without blocking
- randomized complete block design
- factorial designs with outer arrays

- hyper-Graeco Latin square designs

You can also create more complex designs, such as incomplete block designs, by using the FACTEX procedure in conjunction with the DATA step.

You can save the design constructed by the FACTEX procedure in a SAS data set. Once you have run your experiment, you can add the values of the response variable and use the GLM procedure to perform analysis of variance and study significance of effects.

The FACTEX procedure is an interactive procedure. After specifying an initial design, you can submit additional statements without reinvoking the procedure. Once you have constructed a design, you can

- print the design points
- examine the alias structure for the design
- modify the design by changing its size, changing the use of blocking, or respecifying the effects of interest in the model
- output the design to a data set
- examine the confounding rules that generate the design
- randomize the design
- replicate the design
- recode the design from standard values (such as -1 and $+1$) to values appropriate for your situation
- find another design

Learning about the FACTEX Procedure

To learn the basic syntax of the FACTEX procedure, read the "Getting Started" section on page 590, which contains some simple introductory examples. The summary tables in the "Summary of Functions" section on page 601 provide an overview of the syntax. The "Summary of Designs" section on page 603 shows simple ways to construct full factorial designs and fractional factorial designs. The "Advanced Examples" section on page 617 illustrate construction of complex designs.

Getting Started

The following introductory examples illustrate the capabilities of the FACTEX procedure. See "Advanced Examples" on page 617 for illustrations of complex features.

Example of a Two-Level Full Factorial Design

This example FACTEXG1 introduces the basic syntax used with the FACTEX procedure.

An experimenter is interested in studying the effects of three factors—cutting speed (SPEED), feed rate (FEED), and tool angle (ANGLE)—on the surface finish of a metallic part. Two levels of each factor are chosen (shown below), and the experimenter decides to run a complete factorial experiment.

Factor	Low Level	High Level
Cutting Speed	300	500
Feed Rate	20	30
Tool Angle	6	8

This is a 2^3 factorial design—in other words, a complete factorial experiment with three factors, each at two levels. Hence there are eight runs in the experiment. Since complete factorial designs have full resolution, all of the main effects and interaction terms can be estimated. For a definition of the design resolution, see page 651.

You can use the following statements to create the required design:

```
proc factex;                      /* Invoke the FACTEX procedure */
   factors speed feed angle;      /* List factor names           */
   examine design;                /* Display coded design points */
run;
```

By default, the FACTEX procedure assumes that the size of the design is a full factorial and that each factor has only two levels.

After you submit the preceding statements, you will see the following messages in the SAS log:

```
NOTE: No design size specified.
      Default is a full replicate in 8 runs.
NOTE: Design has 8 runs, full resolution.
```

The output is shown in Figure 21.1. The two factor levels are represented by the coded values 1 and -1.

The FACTEX Procedure

Design Points

Experiment Number	speed	feed	angle
1	-1	-1	-1
2	-1	-1	1
3	-1	1	-1
4	-1	1	1
5	1	-1	-1
6	1	-1	1
7	1	1	-1
8	1	1	1

Figure 21.1. 2^3 Factorial Design

If you prefer to work with the actual (decoded) values of the factors, you can specify these values in an OUTPUT OUT= statement, as follows:

```
proc factex;
   factors speed feed angle;
   output out=savdesgn              /* Recode factor levels and  */
          speed nvals=(300 500)     /*   save design in SAVDESGN */
          feed  nvals=(20  30 )
          angle nvals=(6   8  );
proc print;
run;
```

Since the levels in this example are of numeric type, you use the NVALS= option to list the factor levels. Optionally, you can use the CVALS= option for levels of character type (see page 594 for an example). The design is saved in a user-specified output data set (SAVDESGN). This is verified by the following message in the SAS log:

```
NOTE: The data set WORK.SAVDESGN has 8 observations
      and 3 variables.
```

Figure 21.2 shows a listing of the data set SAVDESGN.

Obs	speed	feed	angle
1	300	20	6
2	300	20	8
3	300	30	6
4	300	30	8
5	500	20	6
6	500	20	8
7	500	30	6
8	500	30	8

Figure 21.2. 2^3 Factorial Design after Decoding

Although small complete factorial designs are not difficult to create manually, you can easily extend this example to construct a design with many factors.

Example of a Full Factorial Design in Two Blocks

See FACTEXG2 in the SAS/QC Sample Library

The previous example illustrates a complete factorial experiment involving eight runs and three factors: cutting speed (SPEED), feed rate (FEED), and tool angle (ANGLE).

Now, suppose two machines (A and B) are used to complete the experiment, with four runs being performed on each machine. To allow for the possibility that the machine affects the part finish, you should consider machine as a block factor and account for the block effect in assigning the runs to machines.

The following statements construct a blocked design:

```
proc factex;
    factors speed feed angle;  /* Specify 3 factors, a 2^3 design   */
    blocks nblocks=2;          /* Specify two blocks                */
    model resolution=max;      /* Specify maximum resolution        */
    examine design;
run;
```

The RESOLUTION=MAX option in the MODEL statement specifies a design with the highest resolution, that is, the best design in a general sense. Optionally, if you know the resolution of the design, you can replace RESOLUTION=MAX with RESOLUTION=r where r is the resolution number. For information on resolution, see page 651.

By default, the FACTEX procedure assumes the size of the design is a full factorial and that each factor is at two levels.

After you submit the preceding statements, you will see the following messages in the SAS log:

```
NOTE: No design size specified.
      Default is a full replicate in 8 runs.
NOTE: Design has 8 runs in 2 blocks of size 4,
      resolution = 6.
```

The output is shown in Figure 21.3. Note that, by default, the name for the block variable is BLOCK and its levels are 1 and 2. Also, note that the default factor levels for a two-level design are -1 and 1.

The FACTEX Procedure

Design Points

Experiment Number	speed	feed	angle	Block
1	-1	-1	-1	1
2	-1	-1	1	2
3	-1	1	-1	2
4	-1	1	1	1
5	1	-1	-1	2
6	1	-1	1	1
7	1	1	-1	1
8	1	1	1	2

Figure 21.3. 2^3 Factorial Design in Two Blocks before Decoding

You can rename the block variable and use actual levels for the block variable appropriate for your situation as follows:

```
proc factex;
   factors speed feed angle;
   blocks nblocks=2;
   model resolution=max;
   output out=blocdesn
     speed nvals=(300 500)
     feed  nvals=(20  30 )
     angle nvals=(6   8  )
     blockname=machine cvals=('A' 'B');
run;

proc print;
run;
```

Figure 21.4 shows the listing of the design saved in the data set BLOCDESN.

Obs	machine	speed	feed	angle
1	A	300	20	6
2	A	300	30	8
3	A	500	20	8
4	A	500	30	6
5	B	300	20	8
6	B	300	30	6
7	B	500	20	6
8	B	500	30	8

Figure 21.4. 2^3 Factorial Design in Two Blocks after Decoding

Example of a Half-Fraction Factorial Design

See FACTEX3 in the SAS/QC Sample Library

Often you do not have the resources for a full factorial design. In this case, a fractional factorial design is a reasonable alternative, provided that the effects of interest can be estimated.

Box, Hunter, and Hunter (1978) describe a fractional factorial design for studying a chemical reaction to determine what percentage of the chemicals responded in a reactor. The researchers identified the following five treatment factors that were thought to influence the percentage of reactant:

- the feed rate of the chemicals (FEEDRATE), ranging from 10 to 15 liters per minute
- the percentage of the catalyst (CATALYST), ranging from 1% to 2%
- the agitation rate of the reactor (AGITRATE), ranging from 100 to 120 revolutions per minute
- the temperature (TEMPERAT), ranging from 140 to 180 degrees Centigrade
- the concentration (CONCENTN), ranging from 3% to 6%

The complete 2^5 factorial design requires 32 runs, but it was decided to use a half-fraction design, which requires 16 runs.

Suppose that all main effects and two-factor interactions are to be estimated. An appropriate design for this situation is a design of resolution 5 (denoted as 2^{5-1}_{V}), in which no main effect or two-factor interaction is aliased with any other main effect or two-factor interaction but in which two-factor interactions are aliased with three-factor interactions. This design loses the ability to estimate interactions between three or more factors, but this is usually not a serious loss. For more on resolution, see page 651.

You can use the following statements to construct a 16-run factorial design that has five factors and resolution 5:

```
proc factex;
   factors feedrate catalyst agitrate temperat concentn;
   size design=16;
   model resolution=5;
   output out=reaction feedrate  nvals=(10  15 )
                       catalyst  nvals=(1   2  )
                       agitrate  nvals=(100 120)
                       temperat  nvals=(140 180)
                       concentn  nvals=(3   6  );
proc print;
run;
```

The design saved in the REACTION data set is listed in Figure 21.5.

Obs	feedrate	catalyst	agitrate	temperat	concentn
1	10	1	100	140	6
2	10	1	100	180	3
3	10	1	120	140	3
4	10	1	120	180	6
5	10	2	100	140	3
6	10	2	100	180	6
7	10	2	120	140	6
8	10	2	120	180	3
9	15	1	100	140	3
10	15	1	100	180	6
11	15	1	120	140	6
12	15	1	120	180	3
13	15	2	100	140	6
14	15	2	100	180	3
15	15	2	120	140	3
16	15	2	120	180	6

Figure 21.5. Half-Fraction of a 2^5 Design for Reactors

The use of a half-fraction causes some interaction terms to be confounded with each other. You can use the EXAMINE statement with the ALIASING option to determine which interaction terms are aliased, as follows:

```
proc factex;
   factors feedrate catalyst agitrate temperat concentn;
   size design=16;
   model resolution=5;
   examine aliasing;
run;
```

The alias structure summarizes the estimability of all main effects and two- and three-factor interactions. Figure 21.6 indicates that each of the three-factor interactions is confounded with a two-factor interaction. Thus, if a particular three-factor interaction is believed to be significant, the aliased two-factor interaction cannot be estimated with this half-fraction design.

```
                          The FACTEX Procedure

Aliasing Structure

 feedrate
 catalyst
 agitrate
 temperat
 concentn
 feedrate*catalyst = agitrate*temperat*concentn
 feedrate*agitrate = catalyst*temperat*concentn
 feedrate*temperat = catalyst*agitrate*concentn
 feedrate*concentn = catalyst*agitrate*temperat
 catalyst*agitrate = feedrate*temperat*concentn
 catalyst*temperat = feedrate*agitrate*concentn
 catalyst*concentn = feedrate*agitrate*temperat
 agitrate*temperat = feedrate*catalyst*concentn
 agitrate*concentn = feedrate*catalyst*temperat
 temperat*concentn = feedrate*catalyst*agitrate
```

Figure 21.6. Alias Structure of Reactor Design

When you submit the preceding statements, the following message is displayed in the SAS log:

```
NOTE: Design has 16 runs, resolution = 5.
```

This message confirms that the design exists. If you specify a factorial design that does *not* exist, an error message is displayed in the SAS log. For instance, suppose that you replaced the MODEL statement in the preceding example with the following statement:

```
model resolution=6;
```

Since the maximum resolution of a 2^{5-1} design is 5, the following message appears in the SAS log:

```
ERROR: No such design exists.
```

In general, it is good practice to check the SAS log to see if a design exists.

Using the FACTEX Procedure Interactively

By using the FACTEX procedure interactively, you can quickly explore many design possibilities. The following steps provide one strategy for interactive use:

1. Invoke the procedure with the PROC FACTEX statement, and use a FACTORS statement to identify factors in the design.
2. For a design that involves blocking, use the BLOCKS and MODEL statements. You may want to use the optimization features for the BLOCKS statement.
3. For a fractional replicate of a design, use the SIZE and MODEL statements to specify the characteristics of the design. If the design involves blocking, use a BLOCKS statement as well. If you are unsure of the size of the design or of the number of blocks, use the optimization features for either the BLOCKS or SIZE statement.
4. Enter a RUN statement and check the SAS log to see if the design exists. If a design exists, go on to the next step; otherwise, modify the characteristics given in the SIZE, BLOCKS, and MODEL statements.
5. Examine the alias structure of the design. If it is not appropriate for your situation, go back to step 2 and search for another design.
6. After you have repeated steps 2, 3, and 4 and found an acceptable design, use the OUTPUT statement to save the design. You can optionally recode factor values, recode and rename the block factor, and create new factors using output-value-settings.

Chapter 22
Details of the FACTEX Procedure

Chapter Contents

SYNTAX 601
Summary of Functions 601
Summary of Designs 603

STATEMENT DESCRIPTIONS 605
PROC FACTEX Statement 605
BLOCKS Statement 606
EXAMINE Statement 608
FACTORS Statement 609
MODEL Statement 609
OUTPUT Statement 611
SIZE Statement 615

ADVANCED EXAMPLES 617
Example 22.1. Completely Randomized Design 617
Example 22.2. Resolution IV Augmented Design 618
Example 22.3. Factorial Design with Center Points 620
Example 22.4. Fold-Over Design 621
Example 22.5. Randomized Complete Block Design 623
Example 22.6. Two-Level Design with Design Replication and Point Replication 625
Example 22.7. Mixed-Level Design Using Design Replication and Point Replication 627
Example 22.8. Mixed-Level Design Using Pseudo-Factors 629
Example 22.9. Mixed-Level Design by Collapsing Factors 630
Example 22.10. Hyper-Graeco-Latin Square Design 631
Example 22.11. Resolution IV Design with Minimum Aberration 633
Example 22.12. Replicated Blocked Design with Partial Confounding 635
Example 22.13. Incomplete Block Design 638
Example 22.14. Design with Inner Array and Outer Array 642
Example 22.15. Design and Analysis of a Complete Factorial Experiment . 646

COMPUTATIONAL DETAILS 648
Types of Factors 648
Specifying Effects in the MODEL Statement 649
Factor Variable Characteristics in the Output Data Set 650

STATISTICAL DETAILS 651

Resolution 651
Randomization 652
Replication 654
Confounding Rules 656
Alias Structure 656
Minimum Aberration 657

OUTPUT 658

ODS TABLES 659

Chapter 22
Details of the FACTEX Procedure

Syntax

You can specify the following statements with the FACTEX procedure. Items within the brackets <> are optional.

PROC FACTEX <*options*> **;**
FACTORS *factor-names* < / *option*> **;**
SIZE *size-specification* **;**
MODEL *model-specification* <**MINABS**<*(d)*>>**;**
BLOCKS *block-specification* **;**
EXAMINE <*options*> **;**
OUTPUT **OUT=***SAS-data-set* <*options*> **;**

To generate a design and save it in a data set, you use at least the PROC FACTEX, FACTORS, and OUTPUT statements. The FACTORS statement should immediately follow the PROC FACTEX statement. You use the MODEL and SIZE statements for designs that are less than a full replicate (for example, fractional factorial designs). You can use the BLOCKS statement for designs that involve blocking. The EXAMINE statement can be used as needed.

Summary of Functions

Table 22.1 to Table 22.4 classify the FACTEX statements and options by function.

Table 22.1. Summary of Options for Specifying the Design

Function	Statement	Option
Factor Specification		
Factor names	FACTORS	$factor_1 \ldots factor_f$
Number of levels	FACTORS	$factor_1 \ldots factor_f$ / NLEV=q
Design Size Specification		
(one of the following)		
Number of runs	SIZE	DESIGN=n
Fraction of one full replicate	SIZE	FRACTION=h
Number of *run indexing factors*	SIZE	NRUNFACS=m
Minimum number of runs	SIZE	DESIGN=MINIMUM or FRACTION=MAXIMUM or NRUNFACS=MINIMUM

Function	Statement	Option
Block Specification		
(one of the following)		
Number of blocks	BLOCKS	NBLOCKS=b
Block size	BLOCKS	SIZE=k
Number of *block pseudo-factors*	BLOCKS	NBLKFACS=s
Minimum block size	BLOCKS	NBLOCKS=MAXIMUM or SIZE=MINIMUM or NBLKFACS=MAXIMUM
Model Specification		
(one of the following)		
Estimated effects	MODEL	ESTIMATE=(*effects*)
Estimated effects and non-negligible effects	MODEL	ESTIMATE=(*effects*) NONNEG=(*nonnegligible-effects*)
Design resolution number	MODEL	RESOLUTION=r
Design with highest resolution	MODEL	RESOLUTION=MAXIMUM
Minimum aberration design (up to d^{th} order interactions)	MODEL	EST=(. . .) <NONNEG=(. . .)> or RES=. . . / MINABS<(d)>

Table 22.2. Summary of Options for Searching the Design

Function	Statement	Option
Search for the Design		
Allow maximum time of t seconds	PROC FACTEX	SECONDS=t or TIME=t
Limit the design searches	PROC FACTEX	NOCHECK

Table 22.3. Summary of Options for Replicating and Randomizing the Design

Function	Statement	Option
Replication		
Replicate entire design c times	OUTPUT OUT=*SAS-data-set*	DESIGNREP=c
Replicate design for each point in the data set	OUTPUT OUT=*SAS-data-set*	DESIGNREP=*SAS-data-set*
Replicate each point in design p times	OUTPUT OUT=*SAS-data-set*	POINTREP=p
Replicate data set for each point in the design	OUTPUT OUT=*SAS-data-set*	POINTREP=*SAS-data-set*
Randomization		
Randomize the design	OUTPUT OUT=*SAS-data-set*	RANDOMIZE
Randomize the design but not the assignment of factor levels	OUTPUT OUT=*SAS-data-set*	RANDOMIZE NOVALRAN

Function	Statement	Option
Specify seed number	OUTPUT OUT=*SAS-data-set*	RANDOMIZE (u)

Table 22.4. Summary of Options for Examining and Saving the Design

Function	Statement	Option
List the Design		
Coded factor and block levels	EXAMINE	DESIGN
List the Design Characteristics		
Alias structure (up to d^{th} order interactions)	EXAMINE	ALIASING<(d)>
Confounding rules	EXAMINE	CONFOUNDING
Save the Design		
Coded factor levels	OUTPUT OUT=*SAS-data-set*	
Decoded factor levels (numeric type)	OUTPUT OUT=*SAS-data-set*	*factor-name* NVALS=(*level1 . . . levelq*)
Decoded factor levels (character type)	OUTPUT OUT=*SAS-data-set*	*factor-name* CVALS=(*'level1' . . . 'levelq'*)
Block variable name	OUTPUT OUT=*SAS-data-set*	BLOCKNAME=*block-name*
Decoded block levels (numeric type)	OUTPUT OUT=*SAS-data-set*	BLOCKNAME=*block-name* NVALS=(*level1 . . . levelb*)
Decoded block levels (character type)	OUTPUT OUT=*SAS-data-set*	BLOCKNAME=*block-name* CVALS=(*'level1' . . . 'levelb'*)

Summary of Designs

Table 22.5 summarizes basic design types that you can construct with the FACTEX procedure by providing example code for each type.

Table 22.5. Basic Designs Constructed by the FACTEX Procedure

Design Type	Example Statements
A full factorial design in three factors, each at two levels coded as -1 and $+1$.	**proc factex;** **factors** *pressure temp time* **;** **examine design;** **run;**
A full factorial design in three factors, each at three levels coded as -1, 0, and $+1$.	**proc factex;** **factors** *pressure temp time* **/nlev=** *3* **;** **examine design;** **run;**

Design Type	Example Statements

A full factorial design in three factors, each at two levels. The entire design is replicated twice, and the design with recoded factor levels is saved in a SAS data set.	`proc factex;`
	`  factors` *pressure temp time* `;`
	`  output out=` *savdesgn* `designrep=` *2*
	`  pressure cvals(` *'low' 'high'* `)`
	`  temp nvals(` *200 300* `)`
	`  time nvals(` *10 20* `);`
	`run;`

A full factorial design in three factors, each at two levels coded as -1 and $+1$. Each run in the design is replicated three times, and the replicated design is randomized and saved in a SAS data set.	`proc factex;`
	`  factors` *pressure temp time* `;`
	`  output out=` *savdesgn*
	`  pointrep=`*3* `randomize;`
	`run;`

A full factorial design in three control factors, each at two levels coded as -1 and $+1$. A noise factor design (*outer array*) read from a SAS data set is replicated for each run in the control factor design (*inner array*), and the product design is saved in a SAS data set.	`proc factex;`
	`  factors` *pressure temp time* `;`
	`  output out = ` *savdesgn*
	`  pointrep=` *outarray* `;`
	`run;`

A full factorial blocked design in three factors, each at two levels coded as -1 and $+1$. The design is arranged in two blocks and saved in a SAS data set. By default, the block variable is named BLOCK and the two block levels are numbered 1 and 2.	`proc factex;`
	`  factors` *pressure temp time* `;`
	`  blocks nblocks=` *2* `;`
	`  output out=` *savdesgn* `;`
	`run;`

A full factorial blocked design in three factors, each at two levels coded as -1 and $+1$. Each block contains four runs; the block variable is renamed and the block levels of character type are recoded. The design is saved in a SAS data set.	`proc factex;`
	`  factors` *pressure temp time* `;`
	`  blocks size=` *4* `;`
	`  output out=` *savdesgn*
	`  blockname=` *machine* `cvals(` *'A' 'B'* `);`
	`run;`

Design Type	Example Statements
A fractional factorial design of resolution 4 in four factors, each at two levels coded as -1 and $+1$. The size of the design is eight runs.	**proc factex;** **factors** *pressure temp time catalyst* **;** **size design=** *8* **;** **model resolution=** *4* **;** **examine design;** **run;**
A one-half fraction of a factorial design in four factors, each at two levels coded as -1 and $+1$. The design is of maximum resolution. The design points, the alias structure, and the confounding rules are listed.	**proc factex;** **factors** *pressure temp time catalyst* **;** **size fraction=** *2* **;** **model resolution=maximum;** **examine design aliasing confounding;** **run;**
A one-quarter fraction of a factorial design in six factors, each at two levels coded as -1 and $+1$. Main effects are estimated, and some two-factor interactions are considered nonnegligible. The design is saved in a SAS data set.	**proc factex;** **factors** *x1-x6* **;** **size fraction=** *4* **;** **model estimate=(** *x1 x2 x3 x4 x5 x6* **)** **nonneg =(** *x1*x5 x1*x6 x5*x6* **);** **output out =** *savdesgn* **;** **run;**

Statement Descriptions

This section provides detailed syntax information for the FACTEX procedure statements, beginning with the PROC FACTEX statement. The remaining statements are presented in alphabetical order.

PROC FACTEX Statement

PROC FACTEX <*options*> **;**

You use the PROC FACTEX statement to invoke the FACTEX procedure. The following *options* are available:

NAMELEN
specifies the length of effect names in tables and output data sets to be n characters long, where n is a value between 20 and 200 characters. The default length is 20 characters.

NOCHECK
suppresses a technique for limiting the amount of search required to find a design. The technique dramatically reduces the search time by pruning branches of the search tree that are unlikely to contain the specified design, but in rare cases it can keep the FACTEX procedure from finding a design that does, in fact, exist. The NOCHECK option turns off this technique at the potential cost of an increase in run time. Note, however, that the run time is always bounded by the TIME= option or its default value. For more on the NOCHECK option, see "Speeding Up the Search" on page 667.

TIME=t
SECONDS=t
specifies the maximum number of seconds to spend on the search. The default is 60 seconds.

BLOCKS Statement

BLOCKS *block-specification* **;**
You use the BLOCKS statement to specify the number of blocks in the design or the size of each block in the design. By default, the FACTEX procedure constructs designs that do not contain blocks. If you use the BLOCKS statement, you also need to use the MODEL statement or SIZE statement. In particular, if you use the BLOCKS statement and your design is a fractional factorial design, you must use the MODEL statement.

The two simplest explicit *block-specifications* that you can use are

- NBLOCKS=*b*, which specifies the number of blocks (*b*) in the design
- SIZE=*k*, which specifies the number of runs (*k*) in each block

Use only one of these two options. In all, there are six mutually exclusive *block-specifications* that you can use, as described by the following list:

NBLKFACS=s
specifies the number of *block pseudo-factors* for the design. The design contains a different block for each possible combination of the levels of the block pseudo-factors. Values of s are the integers 1, 2, and so on. See "Block Size Restrictions" on page 607 for details.

If each factor in the design has q levels, then NBLKFACS=s specifies a design with q^s blocks. The size of each block depends on the number of runs in the design, as specified in the SIZE statement. If the design has n runs, then each block has n/q^s runs.

The following statement illustrates how to request a two-level factorial design arranged in eight (2^3) blocks:

```
blocks nblkfacs=3;
```

For more on pseudo-factors, see "Types of Factors" on page 648.

NBLOCKS=b

specifies the number of blocks in the design. The values of b must be a power of q, the number of levels of each factor in the design. See "Block Size Restrictions" on page 607 for details. The size of each block depends on the number of runs in the design, as specified in the SIZE statement. If the design has n runs, then each block has n/b runs. See page 593 for an illustration of this option.

The following statement illustrates how to specify a design arranged in four blocks:

```
blocks nblocks=4;
```

SIZE=k

specifies the number of runs per block in the design. The value k must be a power of q, the number of levels for each factor in the design. The number of blocks depends on the number of runs in the design, as specified in the SIZE statement. If the design has n runs, then it has n/k blocks.

CAUTION: Do not confuse the SIZE= option in the BLOCKS statement with the SIZE statement, which you use to specify the overall size of the design. See page 615 for details of the SIZE statement.

The following statement illustrates how to specify blocks of size two:

```
blocks size=2;
```

NBLKFACS=MAXIMUM
NBLOCKS=MAXIMUM
SIZE=MINIMUM

constructs a blocked design with the minimum number of runs per block, given all the other characteristics of the design. In other words, the block size is optimized. You cannot specify this option if you specify any of the design size optimization options in the SIZE statement (see DESIGN=MINIMUM on page 616).

Equivalence of Specifications

The three explicit *block-specifications* are related to each other, as demonstrated by the following example.

Suppose you want to construct a design for 11 two-level factors in 128 runs in blocks of size 8. Since $128/2^4 = 128/16 = 8$, three equivalent block specifications are

```
blocks nblkfacs=4;
blocks nblocks=16;
blocks size=8;
```

Block Size Restrictions

The number of blocks and the number of runs in each block must be less than the total number of runs in the design. Hence, there are some restrictions on the block size.

- If you use SIZE=k or NBLOCKS=b, the numbers you specify for k and b must be less than or equal to the size of the design, as specified in the SIZE statement.

Or, if you do not use a SIZE statement, k and b must be less than or equal to the number of runs for a full replication of all possible combinations of the factors.

For example, for a 2^3 design you cannot specify a design arranged in 8 blocks (NBLOCKS=8). Likewise, you cannot construct a design with block size greater than 8 (SIZE=8).

- If you use NBLKFACS=s, the value of s can be no greater than the number of *run-indexing factors*, which give the number of runs needed to index the design. For details, see "Types of Factors" on page 648 and Chapter 23, "Theory of Orthogonal Designs," on page 661.

EXAMINE Statement

EXAMINE <*options*> **;**

You use the EXAMINE statement to specify the characteristics of the design that are to be listed in the output.

The *options* are remembered by the procedure; once specified, they remain in effect until you submit a new EXAMINE statement with different options or until you turn off all EXAMINE options by submitting just

```
examine;
```

The following *options* are available.

ALIASING< (d) >
A< (d) >
lists the alias structure of the design, which identifies effects that are confounded with one another and are thus indistinguishable.

You can specify (d) immediately after the ALIASING option for a listing of the alias structure with effects up to and including order d. For example, the following statement requests aliases for up to fourth-order effects (for example, A*B*C*D):

```
examine aliasing(4);
```

Each line of the alias structure is listed in the form

effect=*effect*= . . . =*effect*

for as many effects as are aliased with one another.

The default value for d is determined automatically from the model as follows:

- If you specify the model with a resolution number r in the MODEL statement, then d is the integer part of $(r + 1)/2$.
- If you specify the model with a list of effects in the MODEL statement, then d is the larger of
  - one plus the largest order of an effect to be estimated
  - the largest order of an effect considered to be nonnegligible

where main effects have order 1, two-factor interactions have order 2, and so on. For details on aliasing, see "Alias Structure" on page 656.

CONFOUNDING
C

lists the confounding rules used to construct the design. For the definition of confounding rules, see "Confounding Rules" on page 656 and "Suitable Confounding Rules" on page 664.

DESIGN
D

lists the points in the design in standard order with the factor levels coded. For a description of the randomization and coding rules, see "OUTPUT Statement" on page 611.

FACTORS Statement

FACTORS *factor-names* < / *option*> **;**

You use the FACTORS statement to start the construction of a new design by naming the factors in the design. The FACTORS statement clears all previous specifications for the design (number of runs, block size, and so on). Use it when you want to start a new design. **Note that the FACTORS statement should be the first statement following the PROC FACTEX statement.**

In the FACTORS statement,

factor-names
lists names for the factors in the design. These names must be valid SAS variable names. See "Types of Factors" on page 648 for details.

The following *option* is available:

NLEV=q
specifies the number of levels for each factor in the design. The value of q must be an integer greater than or equal to 2. **The default value for q is 2.** In order to construct a design that involves either fractionation or blocking, q must be either a prime number or an integer power of a prime number. For the reason behind this restriction, see "Structure of General Factorial Designs" on page 663.

MODEL Statement

MODEL *model-specification* <**MINABS** < (d) >> **;**

You use the MODEL statement to provide the model for the construction of the factorial design. The model can be specified either directly by specifying the effects to be estimated with the ESTIMATE= option or indirectly by specifying the resolution of the design with the RESOLUTION= option. **If you create a fractional factorial**

design or if you create a design that involves blocking, the MODEL statement is required.

The two *model-specifications* are described as follows:

ESTIMATE=(*effects***)** <*option*>
identifies the *effects* that you want to estimate with the design. To specify *effects*, simply list the names of main effects, and join terms in interactions with asterisks. The *effects* listed must be enclosed within parentheses. See "Specifying Effects in the MODEL Statement" on page 649 for details. You can use EST or E for the keyword ESTIMATE.

After the ESTIMATE= *option*, you can specify the following *option*:

NONNEGLIGIBLE=(*nonnegligible-effects***)**
identifies nonnegligible effects. These are the effects whose magnitudes are unknown, but you do not necessarily want to estimate them with the design. If you do not want certain effects to be aliased with ESTIMATE= effects, then list them in the NONNEGLIGIBLE= effects. The *nonnegligible-effects* listed must be enclosed within parentheses.

You can use NONNEG or N for the keyword NONNEGLIGIBLE.

For example, suppose that you want to construct a fraction of a 2^4 design in order to estimate the main effects of the four factors. To specify the model, simply list the main effects with the EFFECTS= option, since these are the effects of interest. Furthermore, if you consider the two-factor interactions to be significant but are not interested in estimating them, then list these interactions with the NONNEGLIGIBLE= option.

See Example 22.8 on page 629 for an example using the ESTIMATE= option. See page 661 for details on how the FACTEX procedure interprets the model and derives an appropriate confounding scheme.

RESOLUTION=*r*
RESOLUTION=MAXIMUM
specifies the resolution of the design. The resolution number r must be a positive integer greater than or equal to 3. The interpretation of r is as follows:

- If r is odd, then the effects of interest are taken to be those of order $(r-1)/2$ or less.
- If r is even, then the effects of interest are taken to be those of order $(r-2)/2$ or less, and the nonnegligible effects are taken to be those of order $r/2$ or less.

If you specify RESOLUTION=MAXIMUM, the FACTEX procedure searches for a design with the highest resolution that satisfies the SIZE statement requirements.

You can use RES or R for the keyword RESOLUTION and MAX for MAXIMUM.

For more on design resolution, see "Resolution" on page 651. For an example of *model specification* using the RESOLUTION=r option, see page 595. For an example of the RESOLUTION=MAX option, see page 593.

MINABS $< (d) >$

requests a search for a design that has minimum aberration. Specifying (d) immediately after the MINABS option requests a search for a minimum aberration design involving interactions up to order d. The default value for d is determined automatically from the model as follows:

- If you specify the model with a resolution number r in the MODEL statement, then $d = r + 2$.
- If you specify the model with a list of effects in the MODEL statement, then d is the larger of
  - three plus twice the largest order of an effect to be estimated
  - one plus twice the largest order of an effect considered to be nonnegligible

where main effects have order 1, two-factor interactions have order 2, and so on. See "Minimum Aberration" on page 657 for more information. For an example of the MINABS option, see Example 22.11 on page 633.

Examples of the MODEL Statement

Suppose you specify a design with the following FACTORS statement, where the number of factors f can be replaced with a number:

```
factors x1-xf;
```

Then Table 22.6 lists equivalent ways to specify common models.

Table 22.6. Equivalent of Model Specifications

RES= option	EST= and NONNEG= options
`model res=3`	`model est=(x1-xf);`
`model res=4`	`model est=(x1-xf) nonneg=(x1|x2|x3|...|xf@2);`
`model res=5`	`model est=(x1|x2|x3|...|xf@2);`

The resolution specification is more concise than the effects specification and is also more efficient in an algorithmic sense. To decrease the time required to find a design, particularly for designs with a large number of factors, you should specify your model using the RESOLUTION= option rather than listing the effects. For more information on interpreting the resolution number, see "Resolution" on page 651.

OUTPUT Statement

OUTPUT OUT= *SAS-data-set* <*options*> **;**

You use the OUTPUT statement to save a design in an output data set. Optionally, you can use the OUTPUT statement to modify the design by specifying values to be output for factors, creating new factors, randomizing the design, and replicating the design. You specify the output data set as follows:

OUT=*SAS-data-set*
gives the name of the output data set in which the design is saved. Note that OUT= is required.

options
You can use the *options* to

- recode the values for design factors
- recode the values for the block variable
- replicate the entire design
- replicate each point of the design
- randomize the design
- create derived factors based on the original factors

The following list describes the preceding *options*:

Recode Design Factors

By default, the output data set contains a variable for each factor in the design coded with standard values, as follows:

- For factors with 2 levels ($q = 2$), the values are -1 and +1.
- For factors with 3 levels ($q = 3$), the values are -1, 0, and +1.
- For factors with q levels ($q > 3$), the values are $0, 1, 2, \ldots q - 1$.

You can recode the levels of the factor from the standard levels to levels appropriate for your situation.

For example, suppose that you want to recode a three-level factorial design from the standard levels -1, 0, and +1 to the actual levels. Suppose the factors are pressure (PRESSURE) with character levels, agitation rate (RATE) with numeric levels, and temperature (TEMP) with numeric levels. You can use the following statement to recode the factor levels and save the design in a SAS data set named RECODE:

```
output out=recode pressure cvals=('low' 'medium' 'high')
                  rate     nvals=(20    40       60    )
                  temp     cvals=(100   150      200   );
```

The general form of *options* to recode factors is as follows:

factor-name **NVALS= (***level1 level2 . . . levelq***)**
or

factor-name **CVALS= (***'level1' 'level2' . . . 'levelq'***)**
where

factor-name	gives the name of the design factor.
NVALS=	lists new numeric levels for design factors.
CVALS=	lists new character levels for design factors. Each string can be up to 40 characters long.

When recoding a factor, the NVALS= and CVALS= options map the first value listed to the lowest value for the factor, the second value listed to the next lowest value, and so on. If you rename and recode a factor, the type and length of the new variable are determined by whether you use the CVALS= option (character variable with length equal to the longest string) or the NVALS= option (numeric variable). For more on recoding a factor, see "Factor Variable Characteristics in the Output Data Set" on page 650.

Recode Block Factor

If the design uses blocking, the output data set automatically contains a block variable named BLOCK, and for a design with b blocks, the default values of the block variable are $1, 2, \ldots b$. You can rename the block variable and optionally recode the block levels from the default levels to levels appropriate for your situation.

For example, for a design arranged in four blocks, suppose that the block variable is day of the week (DAY) and that the four block levels of character type are *Mon*, *Tue*, *Wed*, and *Thu*. You can use the following statement to rename the block variable, recode the block levels, and save the design in a SAS data set named RECODE:

```
output out=recode
       blockname=day cvals=('Mon' 'Tue' 'Wed' 'Thu');
```

The general form of *options* to change the block variable name or change the block levels is as follows:

BLOCKNAME= *block-name* <**NVALS= (***level1 level2 . . . levelb***)**>

or

BLOCKNAME= *block-name* <**CVALS= (***'level1' 'level2' . . . 'levelb'***)**>

where

block-name gives a new name for the block factor.

NVALS= lists new numeric levels for the block factor. For details, see "Recode Design Factors" on page 612.

CVALS= lists new character levels for the block factor. For details, see "Recode Design Factors" on page 612.

Note that you can simply rename the block variable using only the BLOCKNAME= option, without using the NVALS= and CVALS= options.

Replicate Entire Design

DESIGNREP=*c*

DESIGNREP=*SAS-data-set*

replicates the entire design. Specify DESIGNREP=c to replicate the design c times, where c is an integer. Alternatively, you can specify a SAS data set with the DESIGNREP option. In this case, the design is replicated once for each point in the DESIGNREP= data set, and the OUT= data set contains the variables in the DESIGNREP= data set as well as the design variables.

In mathematical notation, the OUT= data set is the direct product of the DESIGNREP= data set and the design. If the design is A and the DESIGNREP= data set is B, then the OUT= data set is B $\otimes$ A, where $\otimes$ denotes the direct product.

For details, see "Replication" on page 654. For illustrations of the difference between the DESIGNREP= and POINTREP= options, see Example 22.6 on page 625 and Example 22.7 on page 627.

Replicate Design Point

POINTREP=p
POINTREP=*SAS-data-set*
replicates each point of the design. Specify POINTREP=p to replicate each design point p times, where p is an integer. Alternatively, you can specify a SAS data set with the POINTREP= option. In this case, the POINTREP= data set is replicated once for each point in the design and the OUT= data set contains the variables in the POINTREP= data set as well as the design variables.

In mathematical notation, the OUT= data set is the direct product of the design and the POINT= data set. If the design is A and the POINTREP= data set is B, then the OUT= data set is A $\otimes$ B, where $\otimes$ denotes the direct product.

For details, see "Replication" on page 654. For illustrations of the difference between the DESIGNREP= and POINTREP= options, see Example 22.6 on page 625 and Example 22.7 on page 627.

Randomize Design

RANDOMIZE $<(u)>$ $<$ **NOVALRAN** $>$
randomizes the design. See "Randomization" on page 652 for details. The following *options* are available:

(u)
specifies an integer used to start the pseudo-random number generator for randomizing the design. The value of u must be enclosed in parentheses immediately after the keyword RANDOMIZE. If you don't specify a seed, or specify a value less than or equal to zero, the seed is by default generated from reading the time of day from the computer's clock.

NOVALRAN
prevents the randomization of theoretical factor levels to actual levels. The randomization of run order is still performed.

Create Derived Factors

You can create *derived factors* based on the joint values of a set of the design factors. Each distinct combination of levels of the design factors corresponds to a single level for the derived factor. Thus, when you create a derived factor from k design factors, each with q levels, the derived factor has q^k levels. Derived factors are useful when you create mixed-level designs; see Example 22.8 on page 629 for an example. See "Structure of General Factorial Designs" on page 663 for information on how the

levels of design factors are mapped into levels of the derived factor. The general form of the *option* for creating derived factors is

[*design-factors*]**=** *derived-factor* < **NVALS= (***list-of-numbers***)**>

or

[*design-factors*]**=** *derived-factor* < **CVALS= (***'string1' 'string2' . . . 'stringn'***)**>

where

design-factors	gives names of factors currently in the design. These factors are combined to create the new derived factor.
derived-factor	gives a name to the new derived factor. This name must not be used in the design.
NVALS=	lists new numeric levels for the derived factor.
CVALS=	lists new character levels for the derived factor. See "Recode Design Factors" on page 612 for details.

If you create a derived factor and do not use the NVALS= or CVALS= option to assign levels to the derived factor, the FACTEX procedure assigns the values $0, 1, \ldots, q^k-1$, where the derived factor is created from k design factors, each with q levels. In general, the CVALS= or NVALS= list for a derived factor must contain q^k values.

The following statement gives an example of creating a derived factor and then renaming the levels of the factor:

```
output out=new [a1 a2]=a cvals=('A' 'B' 'C' 'D');
```

This statement converts two two-level factors (A1 and A2) into one four-level factor (A), which has the levels A, B, C, and D.

SIZE Statement

SIZE *size-specification* **;**

You use the SIZE statement to specify the size of the design, which is the number of runs in the design. The SIZE statement is required for designs of less than a full replicate (for example, fractional factorial designs). By default, the design consists of one full replication of all possible combinations of the factors.

The two simplest explicit *size-specifications* that you can use are

- DESIGN=n, which specifies the number of runs (n) in the design
- FRACTION=h, which specifies $1/h$ of one full replicate

Use only one of these two options. In all, there are six mutually exclusive *size-specifications* that you can use, as described by the following list:

DESIGN=n
specifies the actual number of runs in the design. The number of runs must be a power of the number of levels q for the factors in the design. (See the NLEV= option on page 609). If the last FACTORS statement does not contain the NLEV= option, then $q = 2$ by default, and as a result, n must be a power of 2. For an example, see page 617.

FRACTION=h
specifies the fraction of one full replication of all possible combinations of the factors. For instance, FRACTION=2 specifies a half-fraction, and FRACTION=4 specifies a quarter-fraction, and so on. In general, FRACTION=h specifies a design with $1/h$ of the runs in a full replicate. If the design has f factors, each with q levels, then the size of the design is q^f/h. If you use FRACTION=h, h must be a power of q. See Example 22.4 on page 621.

NRUNFACS=m
specifies the number of *run-indexing factors* in the design. The design contains one run for each possible combination of the levels of the run-indexing factors. Run-indexing factors are the first m factors for a design in q^m runs. All possible combinations of the levels of the run-indexing factors occur in the design. As a result, if each factor has q levels, the number of runs in the design is q^m. For details on run-indexing factors, see "Types of Factors" on page 648 and "Structure of General Factorial Designs" on page 663.

DESIGN=MINIMUM
FRACTION=MAXIMUM
NRUNFACS=MINIMUM
constructs a design with the minimum number of runs (no larger than one full replicate) given all of the other characteristics of the design. In other words, the design size is optimized. You cannot specify this option if you specify any of the block size optimization features in the BLOCKS statement (see NBLKFACS=MAXIMUM on page 607).

Equivalence of Specifications

The three explicit *size-specifications* are related to each other, as demonstrated by the following example. Suppose you want to construct a design for 11 two-level factors in 128 runs. Since $128 = 2^{11}/16 = 2^7$, three equivalent size specifications for this design are

```
size design=128;
size fraction=16;
size nrunfacs=7;
```

Advanced Examples

Example 22.1. Completely Randomized Design

See FACTEX8 in the SAS/QC Sample Library

An experimenter wants to study the effect of cutting speed (SPEED) on the surface finish of a component. He considers testing the components at five levels of cutting speed (100, 125, 150, 175, and 200) and decides to test five components at each level.

The design used is a single-factor *completely randomized design* with five levels and 25 runs. The following statements generate the required design:

```
proc factex;
   factors speed / nlev=5;
   size design=25;
   output out=surfexpt randomize        /* Randomly assign run order */
      speed nvals=(100 125 150 175 200);
run;

proc print data=surfexpt;
run;
```

The design saved in the data set SURFEXPT is displayed in Output 22.1.1.

Output 22.1.1. A Completely Randomized Design

Obs	speed
1	150
2	125
3	200
4	100
5	150
6	150
7	100
8	150
9	100
10	200
11	125
12	175
13	100
14	200
15	200
16	175
17	150
18	175
19	175
20	125
21	125
22	200
23	175
24	100
25	125

If you are working through this example on your computer, you might find a different run order in your output. This is due to the difference in the seed value of the random number generator. You can specify a seed value with the RANDOMIZE option. For syntax, see "Randomize Design" on page 614.

Example 22.2. Resolution IV Augmented Design

See RCBD in the SAS/QC Sample Library

Box, Hunter, and Hunter (1978) describe an injection molding experiment involving eight two-level factors: mold temperature (TEMP), moisture content (MOIST), holding pressure (HOLDPR), cavity thickness (THICK), booster pressure (BOOSTPR), cycle time (TIME), screw speed (SPEED), and gate size (GATE).

The design used has 16 runs and is of resolution 4; it is often denoted as 2^{8-4}_{IV}. You can generate this design, shown in Output 22.2.1, with the following statements:

```
proc factex;
   factors temp    moist holdpr thick    /* List factor names        */
           boostpr time  speed  gate;
   size design=16;                       /* Construct 16-run design  */
   model resolution=4;                   /*    of resolution 4       */
   examine design aliasing;              /* List points and aliasing */
run;
```

Output 22.2.1. A 2^{8-4}_{IV} Design

The FACTEX Procedure

Design Points

Experiment Number	temp	moist	holdpr	thick	boostpr	time	speed	gate
1	-1	-1	-1	-1	-1	-1	-1	-1
2	-1	-1	-1	1	1	1	1	-1
3	-1	-1	1	-1	1	1	-1	1
4	-1	-1	1	1	-1	-1	1	1
5	-1	1	-1	-1	1	-1	1	1
6	-1	1	-1	1	-1	1	-1	1
7	-1	1	1	-1	-1	1	1	-1
8	-1	1	1	1	1	-1	-1	-1
9	1	-1	-1	-1	-1	1	1	1
10	1	-1	-1	1	1	-1	-1	1
11	1	-1	1	-1	1	-1	1	-1
12	1	-1	1	1	-1	1	-1	-1
13	1	1	-1	-1	1	1	-1	-1
14	1	1	-1	1	-1	-1	1	-1
15	1	1	1	-1	-1	-1	-1	1
16	1	1	1	1	1	1	1	1

The alias structure is shown in Output 22.2.2.

Output 22.2.2. Alias Structure for a 2^{8-4}_{IV} Design

```
                          The FACTEX Procedure

Aliasing Structure

 temp
 moist
 holdpr
 thick
 boostpr
 time
 speed
 gate
 temp*moist = holdpr*gate = thick*speed = boostpr*time
 temp*holdpr = moist*gate = thick*time = boostpr*speed
 temp*thick = moist*speed = holdpr*time = boostpr*gate
 temp*boostpr = moist*time = holdpr*speed = thick*gate
 temp*time = moist*boostpr = holdpr*thick = speed*gate
 temp*speed = moist*thick = holdpr*boostpr = time*gate
 temp*gate = moist*holdpr = thick*boostpr = time*speed
```

Subsequent analysis of the data collected for the design suggests that HOLDPR and BOOSTPR have statistically significant effects. There also seems to be significant effect associated with the sum of the aliased two-factor interactions TEMP*BOOSTPR, MOIST*TIME, HOLDPR*SPEED, and THICK*GATE. This chain of confounded interactions is identified in Output 22.2.2.

A few runs can be added to the design to distinguish between the effects due to these four interactions. You simply need a design in which any three of these effects are estimable, regardless of all other main effects and interactions. For example, the following statements generate a suitable set of runs (see Output 22.2.3):

```
proc factex;
   factors temp    moist holdpr thick
           boostpr time  speed  gate;
   model estimate=(moist*time
                   holdpr*speed
                   thick*gate  );
   size design=4;
   examine design aliasing(2);
run;
```

Output 22.2.3. Additional Runs to Resolve Ambiguities

The FACTEX Procedure

Design Points

Experiment Number	temp	moist	holdpr	thick	boostpr	time	speed	gate
1	-1	-1	1	1	1	1	-1	1
2	-1	1	-1	-1	-1	-1	-1	1
3	1	-1	-1	-1	-1	-1	1	1
4	1	1	1	1	1	1	1	1

Output 22.2.4 shows the alias structure of the additional four-run experiment. Note that the alias link

TEMP*BOOSTPR = MOIST*TIME = HOLDPR*SPEED = THICK*GATE

found in the original design is broken. When these four runs are added to the original 16 runs, the four two-factor interactions can be estimated separately with the 20 runs.

Output 22.2.4. Alias Structure of the Additional Experiment

The FACTEX Procedure

Aliasing Structure

```
 0 = gate = temp*speed = holdpr*thick = holdpr*boostprs = holdpr*time
   = thick*boostprs = thick*time = boostprs*time
 temp = speed = temp*gate = moist*holdpr = moist*thick = moist*boostprs
      = moist*time = speed*gate
 moist = temp*holdpr = temp*thick = temp*boostprs = temp*time = moist*gate
       = holdpr*speed = thick*speed = boostprs*speed = time*speed
 holdpr = thick = boostprs = time = temp*moist = moist*speed = holdpr*gate
        = thick*gate = boostprs*gate = time*gate
```

Example 22.3. Factorial Design with Center Points

See FACTEX9 in the SAS/QC Sample Library

Factorial designs involving two levels are the most popular experimental designs. For two-level designs, it is assumed that the response is close to linear over the range of the factor levels. To check for curvature and to obtain an independent estimate of error, you can replicate points at the center of a two-level design. Adding center points to the design does not affect the estimates of factorial effects.

To construct a design with center points, you first create a data set with factorial points using the FACTEX procedure and then augment it with center points by using a simple DATA step. The following example illustrates this technique.

A researcher is studying the effect of three two-level factors—current (CURRENT), voltage (VOLTAGE), and time (TIME)—by conducting an experiment using a complete factorial design. The researcher is interested in studying the overall *curvature* over the range of factor levels by adding four center points.

You can construct this design in two stages. First, create the basic 2^3 design with the following statements:

```
proc factex;
   factors current voltage time;
   output out=factdat
          current nvals=(12  28 )
          voltage nvals=(100 200)
          time    nvals=(50  60 );
run;
```

Next, create the center points and append to the basic design as follows:

```
data center(drop=i);
   do i = 1 to 4;
      current = 20;
      voltage = 150;
      time    = 55;
      output;
   end;
data cpdesgn;
   set factdat center;
run;

proc print data=cpdesgn;
run;
```

The design saved in the data set CPDESIGN is displayed in Output 22.3.1. Observations 1 to 8 are the factorial points, and observations 9 to 12 are the center points.

Output 22.3.1. A 2^3 Design with Four Center Points

Obs	current	voltage	time
1	12	100	50
2	12	100	60
3	12	200	50
4	12	200	60
5	28	100	50
6	28	100	60
7	28	200	50
8	28	200	60
9	20	150	55
10	20	150	55
11	20	150	55
12	20	150	55

Example 22.4. Fold-Over Design

See FACTEX10 in the SAS/QC Sample Library

Folding over a fractional factorial design is a method for breaking the links between aliased effects in a design. Folding over a design means adding a new fraction identical to the original fraction except that the signs of all the factors are reversed. The

new fraction is called a *fold-over* design. Combining a fold-over design with the original fraction converts a design of odd resolution r into a design of resolution $r + 1$.* For example, folding over a resolution 3 design yields a resolution 4 design. You can use the FACTEX procedure to construct the original design fraction and a DATA step to generate the fold-over design.

Consider a 1/8 fraction of a 2^6 factorial design with factors A, B, C, D, E, and F. The following statements construct a 2^{6-3}_{III} design:

```
proc factex;
   factors a b c d e f;
   size fraction=8;            /* Specify 1/8 fraction design */
   model resolution=3;         /*     of resolution 3         */
   examine aliasing;
   output out=original;
run;

title 'Original Design';
proc print data=original;
run;
```

The design, which is saved in the data set ORIGINAL, is displayed in Output 22.4.1.

Output 22.4.1. A 2^{6-3}_{III} Design

Original Design

Obs	a	b	c	d	e	f
1	-1	-1	-1	-1	1	1
2	-1	-1	1	1	-1	-1
3	-1	1	-1	1	-1	1
4	-1	1	1	-1	1	-1
5	1	-1	-1	1	1	-1
6	1	-1	1	-1	-1	1
7	1	1	-1	-1	-1	-1
8	1	1	1	1	1	1

Since the design is of resolution 3, the alias structure in Output 22.4.2 indicates that all the main effects are confounded with the two-factor interactions.

*This is not true if the original design has even resolution.

Output 22.4.2. Alias Structure for a 2^{6-3}_{III} Design

```
                    The FACTEX Procedure

Aliasing Structure

 a = c*f = d*e
 b = c*e = d*f
 c = a*f = b*e
 d = a*e = b*f
 e = a*d = b*c
 f = a*c = b*d
 a*b = c*d = e*f
```

To separate the main effects and the two-factor interactions, augment the original design with a 1/8 fraction in which the signs of all the factors are reversed. The combined design (original design and fold-over design) of resolution 4 breaks the alias links between the main effects and the two-factor interactions. The fold-over design can be created using the following DATA step:

```
data foldover;                  /* Create the fold-over design with */
   set original;                /*     the factor signs reversed    */
   a=-a; b=-b; c=-c;
   d=-d; e=-e; f=-f;
run;

title 'Fold-Over Design';
proc print data=foldover;
run;
```

The fold-over design is displayed in Output 22.4.3.

Output 22.4.3. A 2^{6-3}_{III} Design with Signs Reversed

```
                        Fold-Over Design

                Obs     a     b     c     d     e     f

                 1      1     1     1     1    -1    -1
                 2      1     1    -1    -1     1     1
                 3      1    -1     1    -1     1    -1
                 4      1    -1    -1     1    -1     1
                 5     -1     1     1    -1    -1     1
                 6     -1     1    -1     1     1    -1
                 7     -1    -1     1     1     1     1
                 8     -1    -1    -1    -1    -1    -1
```

Example 22.5. Randomized Complete Block Design

In a randomized complete block design (RCBD), each level of a "treatment" appears once in each block, and each block contains all the treatments. The order of

See FACTEX11 in the SAS/QC Sample Library

treatments is randomized separately for each block. You can create RCBDs with the FACTEX procedure.

Suppose you want to construct an RCBD with six treatments in four blocks. To test each treatment once in each block, you need 24 experimental units. The following statements construct the randomized complete block design shown in Output 22.5.1:

```
proc factex;
   factors block / nlev=4;
   output out=blocks
          block nvals=(1 2 3 4);
run;
   factors trt / nlev=6;
   output out=rcbd
          designrep=blocks
          randomize (101)
          trt cvals=('A' 'B' 'C'
                     'D' 'E' 'F');
run;

proc print data=rcbd;
run;
```

Note that the order of the runs within each block is randomized and that the blocks (1, 2, 3, and 4) are in a random order.

Output 22.5.1. A Randomized Complete Block Design

Obs	blocks	trt
1	4	F
2	4	C
3	4	B
4	4	A
5	4	E
6	4	D
7	2	E
8	2	A
9	2	F
10	2	D
11	2	C
12	2	B
13	3	B
14	3	C
15	3	D
16	3	F
17	3	A
18	3	E
19	1	E
20	1	F
21	1	B
22	1	D
23	1	A
24	1	C

Example 22.6. Two-Level Design with Design Replication and Point Replication

See FACTEX12 in the SAS/QC Sample Library

You can replicate a design to obtain an independent estimate of experimental error or to estimate effects more precisely. There are two ways you can replicate a design using the FACTEX procedure: you can replicate the entire design with the DESIGNREP= option, or you can replicate each point in the design with the POINTREP= option. The following example illustrates the difference.

A process engineer is conducting an experiment to study the shrinkage of an injection-molded plastic component. The engineer chooses to determine the effect of the following four factors, each at two levels: holding pressure (PRESSURE), molding temperature (TEMP), cooling time (TIME), and injection velocity (VELOCITY).

The design used is a half-fraction of a 2^4 factorial design, denoted as 2^{4-1}_{IV}. The following statements construct the design:

```
proc factex;
   factors pressure temp time velocity;
   size fraction=2;
   model res=max;
   output out=savunrep;
run;

proc print data=savunrep;
run;
```

The design, saved in the data set SAVUNREP, is shown in Output 22.6.1.

Output 22.6.1. Unreplicated Design

Obs	pressure	temp	time	velocity
1	-1	-1	-1	-1
2	-1	-1	1	1
3	-1	1	-1	1
4	-1	1	1	-1
5	1	-1	-1	1
6	1	-1	1	-1
7	1	1	-1	-1
8	1	1	1	1

To obtain a more precise estimate of the experimental error, the engineer decides to replicate the entire design three times. The following statements generate a 2^{4-1}_{IV} design with three replicates in 24 runs:

```
proc factex;
   factors pressure temp time velocity;
   size fraction=2;
   model res=max;
   output out=savedrep designrep=3;
run;

proc print data=savedrep;
run;
```

The design, saved in the data set SAVEDREP, is displayed in Output 22.6.2.

Output 22.6.2. Design Replication

Obs	pressure	temp	time	velocity
1	-1	-1	-1	-1
2	-1	-1	1	1
3	-1	1	-1	1
4	-1	1	1	-1
5	1	-1	-1	1
6	1	-1	1	-1
7	1	1	-1	-1
8	1	1	1	1
9	-1	-1	-1	-1
10	-1	-1	1	1
11	-1	1	-1	1
12	-1	1	1	-1
13	1	-1	-1	1
14	1	-1	1	-1
15	1	1	-1	-1
16	1	1	1	1
17	-1	-1	-1	-1
18	-1	-1	1	1
19	-1	1	-1	1
20	-1	1	1	-1
21	1	-1	-1	1
22	1	-1	1	-1
23	1	1	-1	-1
24	1	1	1	1

The first replicate comprises observations 1 to 8, the second replicate comprises observations 9 to 16, and the third replicate comprises observations 17 to 24.

Now, instead of replicating the entire design, suppose the engineer decides to replicate each run in the design three times. The following statements construct a 2^{4-1}_{IV} design in 24 runs with point replication:

```
proc factex;
   factors pressure temp time velocity;
   size fraction=2;
   model res=max;
   output out=saveprep pointrep=3;
run;

proc print data=saveprep;
run;
```

The design, saved in the data set SAVEPREP, is displayed in Output 22.6.3. The first design point is replicated three times (observations 1–3), the second design point is replicated three times (observations 4–6), and so on.

Output 22.6.3. Point Replication

Obs	pressure	temp	time	velocity
1	-1	-1	-1	-1
2	-1	-1	-1	-1
3	-1	-1	-1	-1
4	-1	-1	1	1
5	-1	-1	1	1
6	-1	-1	1	1
7	-1	1	-1	1
8	-1	1	-1	1
9	-1	1	-1	1
10	-1	1	1	-1
11	-1	1	1	-1
12	-1	1	1	-1
13	1	-1	-1	1
14	1	-1	-1	1
15	1	-1	-1	1
16	1	-1	1	-1
17	1	-1	1	-1
18	1	-1	1	-1
19	1	1	-1	-1
20	1	1	-1	-1
21	1	1	-1	-1
22	1	1	1	1
23	1	1	1	1
24	1	1	1	1

Note the difference in the arrangement of the designs created using design replication (Output 22.6.2) and point replication (Output 22.6.3). In design replication, the original design is replicated a specified number of times; but in point replication, each run in the original design is replicated a specified number of times. See page 654 for more information on design replication.

Example 22.7. Mixed-Level Design Using Design Replication and Point Replication

See FACTEX13 in the SAS/QC Sample Library

Orthogonal factorial designs are most commonly used at the initial stages of experimentation. At these stages, it is best to experiment with as few levels of each factor as possible in order to minimize the number of runs required. Thus, these designs usually involve only two levels of each factor. Occasionally some factors will naturally have more than two levels of interest—different types of seed, for instance.

You can create designs for factors with different numbers of levels simply by taking the cross product of component designs in which the factors all have the same numbers of levels, that is, replicating every run of one design for each run of the other. (See Example 22.14 on page 642.) All estimable effects in each of the component designs, as well as all generalized interactions between estimable effects in different component designs, are estimable in the cross-product; refer to Section 3 of Chakravarti (1956).

This example illustrates how you can construct a mixed level design using the OUTPUT statement with the POINTREP= option or the DESIGNREP= option to take the cross product between two designs.

Suppose you want to construct a mixed-level factorial design for two two-level factors (A and B) and one three-level factor (C) with 12 runs. The following SAS statements produce a complete 3×2^2 factorial design using design replication:

```
proc factex;
   factors a b;
   output out=ab;
run;
   factors c / nlev=3;
   output out=drepdesn
          designrep=ab;
run;

proc print data=drepdesn;
run;
```

Output 22.7.1 lists the mixed-level design saved in the data set DREPDESN.

Output 22.7.1. 3×2^2 Mixed-Level Design Using Design Replication

Obs	a	b	c
1	-1	-1	-1
2	-1	-1	0
3	-1	-1	1
4	-1	1	-1
5	-1	1	0
6	-1	1	1
7	1	-1	-1
8	1	-1	0
9	1	-1	1
10	1	1	-1
11	1	1	0
12	1	1	1

You can also create a mixed-level design for the preceding factors using the point replication feature of the FACTEX procedure. The following SAS statements produce a complete $2^2 \times 3$ factorial design using point replication:

```
proc factex;
   factors a b;
   output out=ab;
run;
   factors c / nlev=3;
   output out=prepdesn
          pointrep=ab;
run;

proc print data=prepdesn;
run;
```

Output 22.7.2 lists the mixed-level design saved in the data set PREPDESN.

Output 22.7.2. $2^2 \times 3$ Mixed-Level Design Using Point Replication

Obs	c	a	b
1	-1	-1	-1
2	-1	-1	1
3	-1	1	-1
4	-1	1	1
5	0	-1	-1
6	0	-1	1
7	0	1	-1
8	0	1	1
9	1	-1	-1
10	1	-1	1
11	1	1	-1
12	1	1	1

Note the difference between the designs in Output 22.7.1 and Output 22.7.2. In design replication, the mixed-level design is given by AB $\otimes$ C, while for point replication the mixed-level design is given by C $\otimes$ AB, where $\otimes$ denotes the direct product. In design replication, you can view the DESIGNREP= data set as nested *outside* the design, while in point replication, you can view the POINTREP= data set as nested *inside* the design.

Example 22.8. Mixed-Level Design Using Pseudo-Factors

See FACTEX6A in the SAS/QC Sample Library

If the numbers of levels for the factors of the mixed-level design are all powers of the same prime power q, you can construct the design using *pseudo-factors*, where the levels of k q-level pseudo-factors are associated with the levels of a single *derived factor* with q^k levels. Refer to Section 5 of Chakravarti (1956) and see "Types of Factors" on page 648 for details.

For example, the following statements create a design for one four-level factor (A) and three two-level factors (B, C, and D) in 16 runs (a half replicate):

```
proc factex;
   factors a1 a2 b c d;
   model estimate     =(b c d   a1|a2                  )
         nonnegligible=(b|c|d@2 a1|a2|b a1|a2|c a1|a2|d);
   size design=16;
   output out=designa [a1 a2]=a cvals = ('A' 'B' 'C' 'D');
proc print;
  var a b c d;
run;
```

The levels of two two-level pseudo-factors (A1 and A2) are used to represent the four levels of A. Hence the three degrees of freedom associated with A will be given by the main effects of A1 and A2 and their interaction A1*A2, and you can thus refer to (A1|A2) as the main effect of A.

The MODEL statement specifies that the main effects of all factors are to be estimable, and that all of the two-factor interactions between B, C, and D, as well as the interactions between each of these and (A1|A2), are to be nonnegligible. As a result, the mixed-level design has resolution 4. The design is saved in the data set DESIGNA, combining the levels of the two pseudo-factors, A1 and A2, to obtain the levels of the four-level factor A. The data set DESIGNA is listed in Output 22.8.1.

Output 22.8.1. 4×2^3 Design of Resolution IV in 16 Runs

Obs	a	b	c	d
1	A	-1	-1	1
2	A	-1	1	-1
3	A	1	-1	-1
4	A	1	1	1
5	C	-1	-1	-1
6	C	-1	1	1
7	C	1	-1	1
8	C	1	1	-1
9	B	-1	-1	-1
10	B	-1	1	1
11	B	1	-1	1
12	B	1	1	-1
13	D	-1	-1	1
14	D	-1	1	-1
15	D	1	-1	-1
16	D	1	1	1

Example 22.9. Mixed-Level Design by Collapsing Factors

See FACTEX6C in the SAS/QC Sample Library

You can construct a mixed-level design by *collapsing* factors, that is, by replacing a factor with n levels by a factor with m levels, where $m < n$. Orthogonality is retained in the sense that estimates of different effects are uncorrelated, although not all estimates have equal variance; refer to Section 6 of Chakravarti (1956). This method has been used by Addelman (1962) to derive main effects plans for factors with mixed numbers of levels and by Margolin (1967) to construct plans that consider two-factor interactions.

You can use the value specification in the OUTPUT statement as a convenient tool for collapsing factors. For example, the following statements create a 27-run design for two two-level factors (X1 and X2) and two three-level factors (X3 and X4) such that all main effects and two-factor interactions are uncorrelated:

```
proc factex;
   factors x1-x4 / nlev = 3;
   size design=27;
   model r=4;
   output out=savmixed x1 nvals=(-1 1 -1)
                       x2 nvals=(-1 1 -1);
proc print data=savmixed;
run;
```

The mixed-level design is a three-quarter fraction with resolution 5; refer to Margolin (1967). The design is displayed in Output 22.9.1.

Output 22.9.1. $2^2 \times 3^2$ Design of Resolution V in 27 Runs

Obs	x1	x2	x3	x4
1	-1	-1	-1	-1
2	-1	-1	0	1
3	-1	-1	1	0
4	-1	1	-1	1
5	-1	1	0	0
6	-1	1	1	-1
7	-1	-1	-1	0
8	-1	-1	0	-1
9	-1	-1	1	1
10	1	-1	-1	1
11	1	-1	0	0
12	1	-1	1	-1
13	1	1	-1	0
14	1	1	0	-1
15	1	1	1	1
16	1	-1	-1	-1
17	1	-1	0	1
18	1	-1	1	0
19	-1	-1	-1	0
20	-1	-1	0	-1
21	-1	-1	1	1
22	-1	1	-1	-1
23	-1	1	0	1
24	-1	1	1	0
25	-1	-1	-1	1
26	-1	-1	0	0
27	-1	-1	1	-1

Example 22.10. Hyper-Graeco-Latin Square Design

See FACTEX7A in the SAS/QC Sample Library

A $q \times q$ Latin square is an arrangement of q symbols, each repeated q times, in a square of side q such that each symbol appears exactly once in each row and in each column. Such arrangements are useful as designs for *row-and-column* experiments, where it is necessary to balance the effects of two q-level factors simultaneously.

A Graeco-Latin square is actually a pair of Latin squares; when superimposed, each symbol in one square occurs exactly once with each symbol in the other square. The following is an example of a 5×5 Graeco-Latin square, where Latin letters are used for the symbols of one square and Greek letters are used for the symbols of the other:

$$\begin{array}{ccccc} A\alpha & B\beta & C\gamma & D\delta & E\epsilon \\ B\gamma & C\delta & D\epsilon & E\alpha & A\beta \\ C\epsilon & D\alpha & E\beta & A\gamma & B\delta \\ D\beta & E\gamma & A\delta & B\epsilon & C\alpha \\ E\delta & A\epsilon & B\alpha & C\beta & D\gamma \end{array}$$

Whenever q is a power of a prime number, you can construct up to $q - 1$ squares, each with q symbols that are balanced over all the other factors. The result is called a

hyper-Graeco-Latin Square or a complete set of *mutually orthogonal* Latin squares. Such arrangements can be useful as designs (refer to Williams 1949), or they can be used to construct other designs.

When q is a prime power, hyper-Graeco-Latin squares are straightforward to construct with the FACTEX procedure. This is because *a complete set of* $q-1$ *mutually orthogonal* $q \times q$ *Latin squares is equivalent to a resolution 3 design for* $q+1$ q*-level factors in* q^2 *runs, where two of the factors index rows and columns and each of the remaining factors indexes the treatments of one of the squares.*

For instance, the following statements generate a complete set of three mutually orthogonal 4×4 Latin squares, with rows indexed by the factor ROW, columns indexed by the factor COLUMN, and the treatment factors in the respective squares indexed by T1, T2, and T3. The first step is to construct a resolution 3 design for five four-level factors in 16 runs.

```
proc factex;
   factors row column t1-t3 / nlev=4;
   size design=16;
   model resolution=3;
   output out=graeco t1 cvals=('A' 'B' 'C' 'D')
                     t2 cvals=('A' 'B' 'C' 'D')
                     t3 cvals=('A' 'B' 'C' 'D');
run;
```

In most cases, the form that appears in the output data set GRAECO is most useful. The form that usually appears in textbooks is displayed in Output 22.10.1, which can be produced using a simple DATA step (not shown here).

Output 22.10.1. Hyper-Graeco-Latin Square

```
Square 1 :
      A D B C
      D A C B
      B C A D
      C B D A

Square 2 :
      A D B C
      C B D A
      D A C B
      B C A D

Square 3 :
      A D B C
      B C A D
      C B D A
      D A C B
```

Example 22.11. Resolution IV Design with Minimum Aberration

See FACTEX14 in the SAS/QC Sample Library

If a design has resolution IV, then you can simultaneously estimate all main effects and *some* two-factor interactions. However, not all resolution IV designs are equivalent; you may be able to estimate more two-factor interactions with some than with others. Among all resolution IV designs, a design that allows you to estimate the maximum number of two-factor interactions is said to have *minimum aberration*.

For example, if you use the FACTEX procedure to generate a resolution IV two-level design in 32 runs for seven factors, you will be able to estimate all main effects and 15 of the 21 two-factor interactions with the design that is created by default. The following statements create this design and display its alias structure in Output 22.11.1:

```
proc factex;
   factors a b c d e f g;
   model resolution=4;
   size design=32;
   examine aliasing;
run;
```

Output 22.11.1. Alias Structure for Default 2^{7-2}_{IV} Design

```
                    The FACTEX Procedure

Aliasing Structure

 a
 b
 c
 d
 e
 f
 g
 a*b = f*g
 a*c
 a*d
 a*e
 a*f = b*g
 a*g = b*f
 b*c
 b*d
 b*e
 c*d = e*g
 c*e = d*g
 c*f
 c*g = d*e
 d*f
 e*f
```

In contrast, the resolution 4 design given in Table 12.15 of Box, Hunter, and Hunter (1978) is a minimum aberration design that allows estimation of 18 two-factor interactions, three more than can be estimated with the default design. The FACTEX

procedure constructs the minimum aberration design if you specify the MINABS option to the MODEL statement, as in the following statements:

```
proc factex;
   factors a b c d e f g;
   model resolution=4 / minab
   size design=32;
   examine aliasing;
run;
```

The alias structure for the resulting design is shown in Output 22.11.2.

Output 22.11.2. Alias Structure for Minimum Aberration 2^{7-2}_{IV} Design

```
                          The FACTEX Procedure

Aliasing Structure

 a
 b
 c
 d
 e
 f
 g
 a*b
 a*c
 a*d
 a*e
 a*f
 a*g
 b*c
 b*d
 b*e
 b*f
 b*g
 c*d = e*f
 c*e = d*f
 c*f = d*e
 c*g
 d*g
 e*g
 f*g
```

All of the designs listed in Table 12.15 of Box, Hunter, and Hunter (1978) have minimum aberration. For most of these cases, the default design constructed by the FACTEX procedure has minimum aberration—that is, the MINABS option is not required. This is important because the MINABS option forces the FACTEX procedure to check many more designs, and the search can, therefore, take longer to run. You can limit the search time with the TIME= option in the PROC FACTEX statement. In five of the cases (2^{10-6}_{III}, 2^{7-2}_{IV}, 2^{8-3}_{IV}, 2^{9-4}_{IV}, and 2^{10-3}_{V}), the MINABS option is required to construct a design with minimum aberration, and in two cases (2^{9-5}_{III}, 2^{9-3}_{IV}), the NOCHECK option is required as well. If the FACTEX procedure is given a sufficiently large amount of time to run, specifying both the MINABS and the NOCHECK options will always result in a minimum aberration design. However, with the default

search time of 60 seconds, there are three cases (2_{IV}^{10-5}, 2_{IV}^{10-4}, and 2_{IV}^{11-5}) for which the FACTEX procedure is unable to find the minimum aberration design, even with both the MINABS and NOCHECK options specified.

Example 22.12. Replicated Blocked Design with Partial Confounding

See FACTEX15 in the SAS/QC Sample Library

In an unreplicated blocked design, the interaction effect that is confounded with the block effect cannot be estimated. You can replicate the experiment so that a different interaction effect is confounded in each replicate. This enables you to obtain information about an interaction effect from the replicate(s) in which it is not confounded.

For example, consider a 2^3 design with factors A, B, and C arranged in two blocks. Suppose you decide to run four replicates of the design. By constructing the design sequentially, you can choose the effects to be estimated in each replicate depending on the interaction confounded with the block effect in the other replicates.

In the first replicate, you specify only that the main effects are to be estimable. The following statements generate an eight-run two-level design arranged in two blocks:

```
proc factex;
   factors a b c;
   blocks nblocks=2;
   model est=(a b c);
   examine confounding aliasing;
   output out=rep1 blockname=block nvals=(1 2);
run;
```

The alias structure and the confounding scheme are listed in Output 22.12.1. The highest order interaction A*B*C is confounded with the block effect. The design, with recoded block levels, is saved in a dataset named REP1.

Output 22.12.1. Confounding Rule and Alias Structure for Replicate 1

```
                          The FACTEX Procedure

                 Block Pseudo-factor Confounding Rules

                            [B1] = a*b*c



                          The FACTEX Procedure

Aliasing Structure

       a
       b
       c
       a*b
       a*c
       b*c
```

If you were to analyze this replicate by itself, you could not determine whether an effect is due to A*B*C or the block effect. You can construct a second replicate that confounds a different interaction effect with the block effect. Since the FACTEX procedure is interactive, simply submit the following statements to generate the second replicate:

```
      model est=(a b c a*b*c);
      output out=rep2
         blockname=block nvals=(3 4);
  run;
```

The alias structure and the confounding scheme for the second replicate are listed in Output 22.12.2. The interaction A*B*C is free of any aliases, but now the two-factor interaction B*C is confounded with the block effect.

Output 22.12.2. Confounding Rule and Alias Structure for Replicate 2

```
                          The FACTEX Procedure

                 Block Pseudo-factor Confounding Rules

                            [B1] = b*c



                          The FACTEX Procedure

Aliasing Structure

       a
       b
       c
       a*b
       a*c
 [B] = b*c
       a*b*c
```

To estimate the interaction B*C with the third replicate, submit the following statements (immediately after the preceding statements):

```
      model est=(a b c a*b*c b*c);
      output out=rep3 blockname=block nvals=(5 6);
   run;
```

The alias structure and confounding rules are shown in Output 22.12.3. The interaction B*C is free of aliases, but the interaction A*C is confounded with the block effect.

Output 22.12.3. Confounding Rule and Alias Structure for Replicate 3

```
                          The FACTEX Procedure

                  Block Pseudo-factor Confounding Rules

                            [B1] = a*c



                          The FACTEX Procedure

Aliasing Structure

       a
       b
       c
       a*b
 [B] = a*c
       b*c
       a*b*c
```

Finally, to estimate the interaction effect A*C with the fourth replicate, submit the following statements:

```
      model est=(a b c a*b*c b*c a*c);
      output out=rep4 blockname=block nvals=(7 8);
   run;
```

The alias structure and confounding rules are displayed in Output 22.12.4.

Output 22.12.4. Confounding Rule and Alias Structure for Replicate 4

```
                          The FACTEX Procedure

                  Block Pseudo-factor Confounding Rules

                            [B1] = a*b



                          The FACTEX Procedure

Aliasing Structure

       a
       b
       c
 [B] = a*b
       a*c
       b*c
       a*b*c
```

When combined, these four replicates give full information on the main effects and three-quarter information on each of the interactions. The following statements combine the four replicates:

```
data combine;
  set rep1 rep2 rep3 rep4;
run;

proc print data=combine;
run;
```

The final design is saved in the data set COMBINE. A partial listing of this data set is shown in Output 22.12.5.

Output 22.12.5. Combined Design

Obs	block	a	b	c
1	1	-1	-1	-1
2	1	-1	1	1
3	1	1	-1	1
4	1	1	1	-1
5	2	-1	-1	1
6	2	-1	1	-1
7	2	1	-1	-1
8	2	1	1	1
9	3	-1	-1	1
10	3	-1	1	-1
11	3	1	-1	1
12	3	1	1	-1
13	4	-1	-1	-1
14	4	-1	1	1
15	4	1	-1	-1
16	4	1	1	1
17	5	-1	-1	1
18	5	-1	1	1
19	5	1	-1	-1
20	5	1	1	-1
21	6	-1	-1	-1
22	6	-1	1	-1
23	6	1	-1	1
24	6	1	1	1
25	7	-1	1	-1
26	7	-1	1	1
27	7	1	-1	-1
28	7	1	-1	1
29	8	-1	-1	-1
30	8	-1	-1	1
31	8	1	1	-1
32	8	1	1	1

Example 22.13. Incomplete Block Design

See FACTEX7B in the SAS/QC Sample Library

Several important series of balanced incomplete block designs can be derived from orthogonal factorial designs. One is the series on *balanced lattice* of Yates (1936); refer to page 396 of Cochran and Cox (1957). In this situation, the number of treatments v must be the square of a power of a prime number: $v = q^2, \quad q = p^k$ where

p is a prime number. These designs are based on a complete set of $q - 1$ mutually orthogonal $q \times q$ Latin squares, which is equivalent to a resolution 3 design for $q + 1$ q-level factors in q^2 runs.

The balanced lattice designs include $q + 1$ replicates of the treatments. They are constructed by associating each treatment with a run in the factorial design, each replicate with one of the factors, and each block with one of the q values of that factor. For example, the treatments in Block 3 within Replicate 2 are those treatments that are associated with runs for which factor 2 is set at value 3.

The following statements use this method to construct a balanced lattice design for 16 treatments in five replicates of four blocks each. The construction procedure is based on a resolution 3 design for five four-level factors in 16 runs.

```
proc factex;
   factors x1-x5 / nlev=4;
   size design=16;
   model r=3;
   output out=a;
run;
```

In the following DATA step, the incomplete block design is built using the design saved in the data set A by the FACTEX procedure:

```
data b;
   keep rep block plot t;
   array x{5} x1-x5;
   do rep = 1 to 5;
      do block = 1 to 4;
         plot = 0;
         do n = 1 to 16;
            set a point=n;
            if (x{rep}=block-1) then do;
               t = n;
               plot = plot + 1;
               output;
               end;
            end;
         end;
      end;
   stop;
run;
```

For each block within each replicate, the program loops through the run numbers in the factorial design and chooses those which have the REPth factor equal to BLOCK-1. These run numbers are the treatments that go into the particular block.

The design is printed using a DATA step. Each block of each replicate is built into the variables S1, S2, S3, and S4, and each block is printed with a PUT statement.

```
data _null_;
   array s{4} s1-s4;             /* Buffer for holding each block    */
   file print;                   /* Direct printing to output screen */
   n = 1;
   do r = 1 to 5;
      put "Replication " r 1.0 ":";
      do b = 1 to 4;
         do p = 1 to 4;
            set b point=n;
            s{plot} = t;
            n = n+1;
            end;
         put "    Block " b 1.0 ":" (s1-s4) (3.0);
         end;
      put;
      end;
   stop;
run;
```

The design is displayed in Output 22.13.1.

You can use the PLAN procedure to randomize the block design, as shown by the following statements:

```
proc plan seed=54321;
   factors rep=5 block=4 plot=4;
   output data=b out=c;

proc sort;
   by rep block plot;
run;
```

The variable PLOT indexes the plots within each block. Refer to the *SAS/STAT User's Guide* for a general discussion of randomizing block designs.

Finally, substitute **set c** for **set b** in the preceding DATA step. Running this DATA step creates the randomized design displayed in Output 22.13.2.

Output 22.13.1. A Balanced Lattice

```
Replication 1:
     Block 1:  1  2  3  4
     Block 2:  5  6  7  8
     Block 3:  9 10 11 12
     Block 4: 13 14 15 16

Replication 2:
     Block 1:  1  5  9 13
     Block 2:  2  6 10 14
     Block 3:  3  7 11 15
     Block 4:  4  8 12 16

Replication 3:
     Block 1:  1  6 11 16
     Block 2:  3  8  9 14
     Block 3:  4  7 10 13
     Block 4:  2  5 12 15

Replication 4:
     Block 1:  1  8 10 15
     Block 2:  3  6 12 13
     Block 3:  4  5 11 14
     Block 4:  2  7  9 16

Replication 5:
     Block 1:  1  7 12 14
     Block 2:  3  5 10 16
     Block 3:  4  6  9 15
     Block 4:  2  8 11 13
```

Output 22.13.2. Randomized Design

```
Replication 1:
     Block 1: 15  5  2 12
     Block 2:  3  8  9 14
     Block 3: 16  1 11  6
     Block 4:  7 10 13  4

Replication 2:
     Block 1:  2  4  3  1
     Block 2:  5  7  8  6
     Block 3:  9 11 10 12
     Block 4: 15 16 13 14

Replication 3:
     Block 1:  2 13  8 11
     Block 2: 14 12  7  1
     Block 3: 15  4  9  6
     Block 4:  5 16  3 10

Replication 4:
     Block 1: 13  1  5  9
     Block 2: 14  2 10  6
     Block 3: 11 15  3  7
     Block 4: 16 12  4  8

Replication 5:
     Block 1:  2 16  7  9
     Block 2: 15 10  8  1
     Block 3:  3 12  6 13
     Block 4:  5 11 14  4
```

Example 22.14. Design with Inner Array and Outer Array

See FACTEX4 in the SAS/QC Sample Library

Byrne and Taguchi (1986) report the use of a fractional factorial design to investigate fitting an elastomeric connector to a nylon tube as tightly as possible. Their experiment applies the design philosophy of Genichi Taguchi, which distinguishes between *control factors* and *noise factors*. Control factors are typically those that the engineer is able to set under real conditions, while noise factors vary uncontrollably in practice (though within a predictable range).

The experimental layout consists of two designs, one for the control factors and one for the noise factors. The design for the control factors is called the *inner array*, and the design for noise factors is called the *outer array*. The outer array is replicated for each of the runs in the inner array, and a performance measure ("signal-to-noise ratio") is computed over the replicate. The performance measure thus reflects variation due to changes in the noise factors. You can construct such a cross-product design with the replication options in the OUTPUT statement of the FACTEX procedure, as shown in this example.

Researchers identified the following four control factors that were thought to influence the amount of force required to pull the connector off the tube:

- the interference (INTERFER), defined as the difference between the outer width of the tubing and the inner width of the connector
- the connector wall thickness (CONNWALL)
- the depth of insertion (IDEPTH) of the tubing into the connector
- the amount of adhesive (GLUE) in the connector pre-dip

Researchers also identified the following three noise factors related to the assembly:

- the amount of time (TIME) allowed for assembly
- the temperature (TEMPERAT)
- the relative humidity (HUMIDITY)

Three levels were selected for each of the control factors, and two levels were selected for each of the noise factors.

The following statements construct the 72-run design used by Byrne and Taguchi (1986). First, an eight-run outer array for the three noise factors is created and saved in the data set OUTERARY.

```
proc factex;
   factors time temperat humidity;
   output out=outerary time     nvals=( 24 120)
                        temperat nvals=( 72 150)
                        humidity nvals=(.25 .75);
run;
```

Next, a nine-run inner array (design of resolution 3) is chosen for the control factors. The POINTREP option in the OUTPUT statement replicates the eight-run outer array in the data set OUTERARY for each of the nine runs in the inner array and saves the final design containing 72 runs in the data set SAVEDESN.

```
proc factex;
   factors interfer connwall idepth glue / nlev=3;
   size design=9;
   model resolution=3;
   output out=savedesn pointrep=outerary
        interfer cvals=('Low'     'Medium' 'High'  )
        connwall cvals=('Thin'    'Medium' 'Thick' )
        idepth   cvals=('Shallow' 'Deep'   'Medium')
        glue     cvals=('Low'     'High'   'Medium');
run;

proc print data=savedesn;
run;
```

The final design is listed in Output 22.14.1. Main effects of each factor can be estimated free of each other but are confounded with two-factor interactions.

Output 22.14.1. Design for Control Factor and Noise Factors

Obs	interfer	connwall	idepth	glue	time	temperat	humidity
1	Low	Thin	Shallow	Low	24	72	0.25
2	Low	Thin	Shallow	Low	24	72	0.75
3	Low	Thin	Shallow	Low	24	150	0.25
4	Low	Thin	Shallow	Low	24	150	0.75
5	Low	Thin	Shallow	Low	120	72	0.25
6	Low	Thin	Shallow	Low	120	72	0.75
7	Low	Thin	Shallow	Low	120	150	0.25
8	Low	Thin	Shallow	Low	120	150	0.75
9	Low	Medium	Medium	Medium	24	72	0.25
10	Low	Medium	Medium	Medium	24	72	0.75
11	Low	Medium	Medium	Medium	24	150	0.25
12	Low	Medium	Medium	Medium	24	150	0.75
13	Low	Medium	Medium	Medium	120	72	0.25
14	Low	Medium	Medium	Medium	120	72	0.75
15	Low	Medium	Medium	Medium	120	150	0.25
16	Low	Medium	Medium	Medium	120	150	0.75
17	Low	Thick	Deep	High	24	72	0.25
18	Low	Thick	Deep	High	24	72	0.75
19	Low	Thick	Deep	High	24	150	0.25
20	Low	Thick	Deep	High	24	150	0.75
21	Low	Thick	Deep	High	120	72	0.25
22	Low	Thick	Deep	High	120	72	0.75
23	Low	Thick	Deep	High	120	150	0.25
24	Low	Thick	Deep	High	120	150	0.75
25	Medium	Thin	Medium	High	24	72	0.25
26	Medium	Thin	Medium	High	24	72	0.75
27	Medium	Thin	Medium	High	24	150	0.25
28	Medium	Thin	Medium	High	24	150	0.75
29	Medium	Thin	Medium	High	120	72	0.25
30	Medium	Thin	Medium	High	120	72	0.75

Output 22.14.1. (continued)

Obs	interfer	connwall	idepth	glue	time	temperat	humidity
31	Medium	Thin	Medium	High	120	150	0.25
32	Medium	Thin	Medium	High	120	150	0.75
33	Medium	Medium	Deep	Low	24	72	0.25
34	Medium	Medium	Deep	Low	24	72	0.75
35	Medium	Medium	Deep	Low	24	150	0.25
36	Medium	Medium	Deep	Low	24	150	0.75
37	Medium	Medium	Deep	Low	120	72	0.25
38	Medium	Medium	Deep	Low	120	72	0.75
39	Medium	Medium	Deep	Low	120	150	0.25
40	Medium	Medium	Deep	Low	120	150	0.75
41	Medium	Thick	Shallow	Medium	24	72	0.25
42	Medium	Thick	Shallow	Medium	24	72	0.75
43	Medium	Thick	Shallow	Medium	24	150	0.25
44	Medium	Thick	Shallow	Medium	24	150	0.75
45	Medium	Thick	Shallow	Medium	120	72	0.25
46	Medium	Thick	Shallow	Medium	120	72	0.75
47	Medium	Thick	Shallow	Medium	120	150	0.25
48	Medium	Thick	Shallow	Medium	120	150	0.75
49	High	Thin	Deep	Medium	24	72	0.25
50	High	Thin	Deep	Medium	24	72	0.75
51	High	Thin	Deep	Medium	24	150	0.25
52	High	Thin	Deep	Medium	24	150	0.75
53	High	Thin	Deep	Medium	120	72	0.25
54	High	Thin	Deep	Medium	120	72	0.75
55	High	Thin	Deep	Medium	120	150	0.25
56	High	Thin	Deep	Medium	120	150	0.75
57	High	Medium	Shallow	High	24	72	0.25
58	High	Medium	Shallow	High	24	72	0.75
59	High	Medium	Shallow	High	24	150	0.25
60	High	Medium	Shallow	High	24	150	0.75
61	High	Medium	Shallow	High	120	72	0.25
62	High	Medium	Shallow	High	120	72	0.75
63	High	Medium	Shallow	High	120	150	0.25
64	High	Medium	Shallow	High	120	150	0.75
65	High	Thick	Medium	Low	24	72	0.25
66	High	Thick	Medium	Low	24	72	0.75
67	High	Thick	Medium	Low	24	150	0.25
68	High	Thick	Medium	Low	24	150	0.75
69	High	Thick	Medium	Low	120	72	0.25
70	High	Thick	Medium	Low	120	72	0.75
71	High	Thick	Medium	Low	120	150	0.25
72	High	Thick	Medium	Low	120	150	0.75

Note that the levels of IDEPTH and GLUE are listed in the OUTPUT statement in a nonstandard order so that the design produced by the FACTEX procedure matches the design of Byrne and Taguchi (1986). The order of assignment of levels does not affect the properties of the resulting design. Furthermore, design can be randomized with the RANDOMIZE option in the OUTPUT statement.

Byrne and Taguchi (1986) indicate that a smaller outer array with only four runs would have been sufficient. You can generate this design (not shown here) by modifying the statements on page 642; specifically, add the following SIZE and MODEL statements:

```
size design=4;
model resolution=3;
```

In their analysis of the data from the experiment based on the smaller design, Byrne and Taguchi (1986) note several interesting interactions between control and noise factors. However, since the inner array is of resolution 3, it is impossible to say whether or not there exist interesting interactions between the control factors. In other words, you cannot determine whether an effect is due to an interaction or to the main effect with which it is confounded.

One alternative is to begin with a design of resolution 4. Two-factor interactions will remain confounded with one another, but they will be free of main effects. Moreover, further experimentation can be carried out to distinguish between confounded interactions that seem important. To determine the optimal size of this design, submit the following statements interactively:

```
proc factex;
   factors interfer connwall idepth glue / nlev=3;
   model resolution=4;
   size design=minimum;
run;
```

This causes the following message to appear in the SAS log:

```
NOTE: Design has 27 runs, resolution = 4.
```

In other words, the smallest resolution 4 design for four three-level factors has 27 runs, which together with the eight-run outer array requires 216 runs. Even the smaller four-run outer array requires 108 runs. Both of these designs are substantially larger than the design originally reported, but the larger designs protect against the effects of unsuspected interactions.

A second alternative is to begin with only two levels of the control factors. Further experimentation can then be directed toward exploring the effects of factors determined to be important in this initial stage of experimentation. Note that NLEV=2 is the default in the FACTORS statement. Submit the following additional statements:

```
   factors interfer connwall idepth glue;
   model resolution=4;
   size design=minimum;
run;
```

This causes the following message to appear in the SAS log:

```
NOTE: Design has 8 runs, resolution = 4.
```

Thus, as few as eight runs can be used for the inner array. This design is amenable to blocking, whereas the proposed nine-run design is not. Blocking is an important consideration whenever experimental conditions can vary over the course of conducting the experiment.

Now, submit the following statements:

```
   size design=8;
   blocks size=minimum;
run;
```

This causes the following message to appear in the SAS log:

```
NOTE: Design has 8 runs in 4 blocks of size 2,
      resolution = 4.
```

Thus the experiment can be run in blocks as small as two runs.

Example 22.15. Design and Analysis of a Complete Factorial Experiment

See FACTEX16 in the SAS/QC Sample Library

Yin and Jillie (1987) describe an experiment on a nitride etch process for a single wafer plasma etcher. The experiment was run using four factors: cathode power (POWER), gas flow (FLOW), reactor chamber pressure (PRESSURE), and electrode gap (GAP). A single replicate of a 2^4 design was run, and the etch rate (RATE) was measured.

You can use the following statements to construct a 16-run design in the four factors:

```
proc factex;
   factors power flow pressure gap;
   output out=desgndat
          power    nvals=(0.80 1.20)
          flow     nvals=(4.50 550 )
          pressure nvals=(125  200 )
          gap      nvals=(275  325 );
run;
```

The design with the actual (decoded) factor levels is saved in the data set DESGNDAT. The experiment using the 16-run design is performed, and the etch rate is measured. The following DATA step updates the data set DESGNDAT with the values of RATE:

```
data desgndat;
   set desgndat;
   input rate @@;
   datalines;
   550   669  604   650  633   642  601   635
   1037  749  1052  868  1075  860  1063  729
   ;
```

The data set DESGNDAT is listed in Output 22.15.1.

Output 22.15.1. A 2^4 Design with Responses

Nitride Etch Process Experiment

Obs	power	flow	pressure	gap	rate
1	0.8	4.5	125	275	550
2	0.8	4.5	125	325	669
3	0.8	4.5	200	275	604
4	0.8	4.5	200	325	650
5	0.8	550.0	125	275	633
6	0.8	550.0	125	325	642
7	0.8	550.0	200	275	601
8	0.8	550.0	200	325	635
9	1.2	4.5	125	275	1037
10	1.2	4.5	125	325	749
11	1.2	4.5	200	275	1052
12	1.2	4.5	200	325	868
13	1.2	550.0	125	275	1075
14	1.2	550.0	125	325	860
15	1.2	550.0	200	275	1063
16	1.2	550.0	200	325	729

To perform an analysis of variance on the responses, you can use the GLM procedure, as follows:

```
proc glm data=desgndat;
   class power flow pressure gap;
   model rate=power|flow|pressure|gap@2 / ss1;
run;
```

The factors are listed in both the CLASS and MODEL statements, and the response as a function of the factors is modeled using the MODEL statement. The MODEL statement requests Type I sum of squares (SS1) and lists all effects that contain two or fewer factors. It is assumed that three-factor and higher interactions are not significant.

Part of the output from the GLM procedure is shown in Output 22.15.2. The main effect of the factors POWER and GAP and the interaction between POWER and GAP are significant (their p-values are less than 0.01).

Output 22.15.2. Analysis of Variance for the Nitride Etch Process Experiment

The GLM Procedure

Dependent Variable: rate

Source	DF	Type I SS	Mean Square	F Value	Pr > F
power	1	374850.0625	374850.0625	183.99	<.0001
flow	1	217.5625	217.5625	0.11	0.7571
power*flow	1	18.0625	18.0625	0.01	0.9286
pressure	1	10.5625	10.5625	0.01	0.9454
power*pressure	1	1.5625	1.5625	0.00	0.9790
flow*pressure	1	7700.0625	7700.0625	3.78	0.1095
gap	1	41310.5625	41310.5625	20.28	0.0064
power*gap	1	94402.5625	94402.5625	46.34	0.0010
flow*gap	1	2475.0625	2475.0625	1.21	0.3206
pressure*gap	1	248.0625	248.0625	0.12	0.7414

Source	DF	Type III SS	Mean Square	F Value	Pr > F
power	1	374850.0625	374850.0625	183.99	<.0001
flow	1	217.5625	217.5625	0.11	0.7571
power*flow	1	18.0625	18.0625	0.01	0.9286
pressure	1	10.5625	10.5625	0.01	0.9454
power*pressure	1	1.5625	1.5625	0.00	0.9790
flow*pressure	1	7700.0625	7700.0625	3.78	0.1095
gap	1	41310.5625	41310.5625	20.28	0.0064
power*gap	1	94402.5625	94402.5625	46.34	0.0010
flow*gap	1	2475.0625	2475.0625	1.21	0.3206
pressure*gap	1	248.0625	248.0625	0.12	0.7414

Computational Details

Types of Factors

The *factors* of a design are variables that an experimenter can set at several values. In general, experiments are performed to study the effects of different levels of the factors on the *response* of interest. For example, consider an experiment to maximize the percentage of raw material that responds to a chemical reaction. The factors might include the reaction temperature and the feed rate of the chemicals, while the response is the yield rate. Factors of different types are used in different ways in constructing a design. This section defines the different types of factors.

Block factors are unavoidable factors that are known to affect the response, but in a relatively uninteresting way. For example, in the chemical experiment, the technician operating the equipment might have a noticeable effect on the yield of the process. The operator effect might be unavoidable, but it is usually not very interesting. On the other hand, factors whose effects are directly of interest are called *design factors*. One goal in designing an experiment is to avoid getting the effects of the design factors mixed up, or *confounded*, with the effects of any block factors.

When constructing a design by orthogonal confounding, all factors formally have the same number of levels q, where q is a prime number or a power of a prime number.

Usually, q is two, and the factor levels are chosen to represent high and low values.

However, this does not mean, for example, that a design for two-level factors is restricted to no more than two blocks. Instead, the values of several two-level factors can be used to index the values of a single factor with more than two levels. As an example, the values of three two-level factors (P_1, P_2, and P_3) can be used to index the values of an eight-level factor(F), as follows:

P_1	P_2	P_3	F
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

The factors P_i are used only to derive the levels of the factor F; thus, they are called *pseudo-factors*, and F is called a *derived factor*. In general, k q-level pseudo-factors give rise to a single q^k-level derived factor. Block factors can be derived factors, and their associated formal factors (the P_i factors) are called *block pseudo-factors*.

The method for constructing an orthogonally confounded design for q-level factors in q^m runs distinguishes between the first m factors and the remaining factors. Each of the q^m different combinations of the first m factors occurs once in the design in an order similar to the preceding table. For this reason, the first m factors are called the *run-indexing factors*.

Table 22.7 summarizes the different types of factors discussed in this section.

Table 22.7. Types of Factors

Term	Definition
Block factor	Unavoidable factor whose effect is not of direct interest
Block pseudo-factor	Pseudo-factor used to derive levels of a block factor
Derived factor	Factor whose levels are derived from pseudo-factors
Design factor	Factor whose effect is of direct interest
Pseudo-factor	Formal factor combined to derive the levels of a real factor
Run-indexing factors	The first m design factors, whose q^m combinations index the runs in the design

Specifying Effects in the MODEL Statement

The FACTEX procedure accepts models that contain terms for main effects and interactions. *Main effects* are specified by writing variable names by themselves.

```
A B C
```

Interactions are specified by joining variable names with asterisks.

```
A*B B*C A*B*C
```

In addition, the *bar operator* (|) simplifies specification for interactions. The `@` *operator*, used in combination with the bar operator, further simplifies specification of interactions. For example, two ways of writing the complete set of effects for a model with up to three-factor interactions are

```
model estimate=(a b c a*b a*c b*c a*b*c);
```

and

```
model estimate=(a|b|c);
```

When the bar (|) is used, the right- and left-hand sides become effects, and their cross becomes an interaction effect. Multiple bars are permitted. The expressions are expanded from left to right, using rules given by Searle (1971). For example, `A|B|C` is evaluated as follows:

```
A | B | C  →  { A | B } | C
           →  { A B A*B } | C
           →  A  B  A*B  C  A*C  B*C  A*B*C
```

You can also specify the maximum number of variables involved in any effect that results from bar evaluation by specifying the number, preceded by an @ sign, at the end of the bar effect. For example, the specification `A|B|C@2` results in only those effects that contain two or fewer factors. In this case, the effects A, B, A*B, C, A*C, and B*C are generated.

Factor Variable Characteristics in the Output Data Set

When you use the OUTPUT statement to save a design in a data set, and you rename and recode a factor, the type and length of the new variable are determined by whether you use the NVALS= or CVALS= option. A factor variable whose values are coded with the NVALS= specification is of numeric type. A factor variable whose values are coded with the CVALS= option is of character type, and the length of the variable is set to the length of the longest character string; shorter strings are padded with trailing blanks.

For example, in the specifications

```
cvals=('String 1' 'A longer string')
cvals=('String 1' 'String 2')
```

the first value in the first CVALS= specification is padded with seven trailing blanks. One consequence is that it no longer matches the 'String 1' of the second specification. To match two such values (for example, when merging two designs), use the TRIM function in the DATA step (see *SAS Language Reference: Dictionary* for details).

Statistical Details

Resolution

The resolution of a design indicates which effects can be estimated free of other effects. The resolution of a design is generally defined as the smallest *order*[*] of the interactions that are confounded with zero. Since having an effect of order $n + m$ confounded with zero is equivalent to having an effect of order n confounded with an effect of order m, the resolution can be interpreted as follows:

- If r is odd, then effects of order $e = (r - 1)/2$ or less can be estimated free of each other. However, at least some of the effects of order e are confounded with interactions of order $e + 1$. A design of odd resolution is appropriate when effects of interest are those of order e or less, while those of order $e + 1$ or higher are all negligible.
- If r is even, then effects of order $e = (r - 2)/2$ or less can be estimated free of each other and are also free of interactions of order $e + 1$. A design of even resolution is appropriate when effects of order e or less are of interest, effects of order $e + 1$ are not negligible, and effects of order $e + 2$ or higher are negligible. If the design uses blocking, interactions of order $e + 1$ or higher may be confounded with blocks.

In particular, for resolution 5 designs, all main effects and two-factor interactions can be estimated free of each other. For resolution 4 designs, all main effects can be estimated free of each other and free of two-factor interactions, but some two-factor interactions are confounded with each other and/or with blocks. For resolution 3 designs, all main effects can be estimated free of each other, but some of them are confounded with two-factor interactions.

In general, higher resolutions require larger designs. Resolution 3 designs are popular because they handle relatively many factors in a minimal number of runs. However, they offer no protection against interactions. If resources allow, you should use a resolution 5 design so that all main effects and two-factor interactions will be independently estimable. If a resolution 5 design is too large, you should use a design of resolution 4, which ensures estimability of main effects free of any two-factor interactions. In this case, if data from the initial design reveal significant effects associated with confounded two-factor interactions, further experiments can be run to distinguish between effects that are confounded with each other in the design. See page 618 for an example.

[*]The order of an effect is the number of factors involved in it. For example, main effects have order one, two-factor interactions have order two, and so on.

Note that most references on fractional factorial designs use Roman numerals to denote resolution of a design—III, IV, V, and so on. A common notation for an orthogonally confounded design of resolution r for k q-level factors in q^{k-p} runs is

$$q_r^{k-p}$$

For example, 2_{V}^{5-1} denotes a design for five two-level factors in 16 runs that allows estimation of all main effects and two-factor interactions. This chapter uses Arabic numerals for resolution since these are specified with the RESOLUTION= option in the MODEL statement.

Randomization

In many experiments, proper randomization is crucial to the validity of the conclusions. Randomization neutralizes the effects of systematic biases that may be involved in implementing the design and provides a basis for the assumptions underlying the analysis. Refer to Kempthorne (1975) for a discussion.

The way in which randomization is handled depends on whether the design involves blocking.

- For designs without block factors, proper randomization consists of randomly permuting the overall order of the runs and randomly assigning the actual levels of each factor to the theoretical levels it has for the purpose of constructing the design.
- For designs with block factors, proper randomization calls for first performing separate random permutations for the runs within each block, and then randomly permuting the order in which the blocks are run.

For example, suppose you generate a full factorial design for three two-level factors A, B, and C in eight runs. The following steps are involved in randomizing this design:

1. Randomly permute the order of the runs.

$$\text{Runs: } \{1,2,3,4,5,6,7,8\} \quad \rightarrow \quad \{3,8,1,2,4,7,6,5\}$$

2. Randomly assign the actual levels to the theoretical levels for each factor.

$$\begin{aligned}\text{Factor A levels: } \{0,1\} &\rightarrow \{1,-1\}\\ \text{Factor B levels: } \{0,1\} &\rightarrow \{1,-1\}\\ \text{Factor C levels: } \{0,1\} &\rightarrow \{-1,1\}\end{aligned}$$

Thus, the effect of the randomization is to transform the original design, as follows:

Run	A	B	C
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

$\longrightarrow$

Run	A	B	C
3	1	-1	-1
8	-1	-1	1
1	1	1	-1
2	1	1	1
4	1	-1	1
7	-1	-1	-1
6	-1	1	1
5	-1	1	-1

If the original design is in two blocks, then the first step is replaced with the following:

1. Randomly permute the order of the runs within each block.

   Block 1 runs: $\{1, 2, 3, 4\} \rightarrow \{4, 1, 2, 3\}$
   Block 2 runs: $\{5, 6, 7, 8\} \rightarrow \{8, 7, 6, 5\}$

2. Randomly permute the order of the blocks.

   Block levels: $\{1, 2\} \rightarrow \{2, 1\}$

The resulting transformation is shown in the following:

Run	Block	A	B	C
1	1	0	0	0
2	1	0	1	1
3	1	1	0	1
4	1	1	1	0
5	2	0	0	1
6	2	0	1	0
7	2	1	0	0
8	2	1	1	1

$\longrightarrow$

Run	Block	A	B	C
8	2	-1	-1	1
7	2	-1	1	-1
6	2	1	-1	-1
5	2	1	1	1
4	1	-1	-1	-1
1	1	1	1	-1
2	1	1	-1	1
3	1	-1	1	1

If you use the RANDOMIZE option in the OUTPUT statement, the output data set contains a randomized design. In some cases, it is appropriate to randomize the run order but not the assignment of theoretical factor levels to actual levels. In these cases, specify both the NOVALRAN and RANDOMIZE options in the OUTPUT statement.

Replication

In quality improvement applications, it is often important to analyze both the mean response of a process and the variability around the mean. To study variability with an experimental design, you must take several measurements of the response for each different combination of the factors of interest; that is, you must *replicate* the design runs.

Replicating a Fixed Number of Times

A simple method of replication is to take a given number of measurements for each combination of factor levels in the basic design. You can replicate runs in the design by specifying numbers for the POINTREP= and DESIGNREP= options in the OUTPUT statement. For example, the following code constructs a full 2^2 design and uses both of these options to replicate the design three times:

```
proc factex;
   factors a b;
   output out=one pointrep =3;
run;
   output out=two designrep=3;
run;
```

The output data sets ONE and TWO have the same 12 runs, but they are in different orders. In the data set ONE, the POINTREP= option causes all three replications of each run to occur together, as shown in Figure 22.1.

OBS	A	B	
1	−1	−1	*3 replicates of run 1*
2	−1	−1	
3	−1	−1	
4	−1	1	*3 replicates of run 2*
5	−1	1	
6	−1	1	
7	1	−1	*3 replicates of run 3*
8	1	−1	
9	1	−1	
10	1	1	*3 replicates of run 4*
11	1	1	
12	1	1	

Figure 22.1. Four-Run Design Replicated Using the POINTREP= Option

On the other hand, in the data set TWO, the DESIGNREP= option causes all four runs of the design to occur together three times, as shown in Figure 22.2.

	OBS	A	B
Replicate 1	1	−1	−1
	2	−1	1
	3	1	−1
	4	1	1
Replicate 2	5	−1	−1
	6	−1	1
	7	1	−1
	8	1	1
Replicate 3	9	−1	−1
	10	−1	1
	11	1	−1
	12	1	1

Figure 22.2. Four-Run Design Replicated Using the DESIGNREP= Option

Replicating with an Outer Array

Another method of design replication considers the range of environmental conditions over which the process should maintain consistency. This method distinguishes between *control factors* and *noise factors*. Control factors are factors that are under the control of the designer or the process engineer. Noise factors cause the performance of a product to vary when the nominal values of the control variables are fixed (noise factors are controllable for the purposes of experimenting with the process). Typical noise factors are variations in the manufacturing environment or the customer's environment due to temperature or humidity. The object of experimentation is to find the best settings for the control factors for a variety of settings for the noise factors. In other words, the goal is to develop a process that runs well in a variety of environments. Refer to Dehnad (1989) and Phadke (1989) for further discussion.

To achieve this goal, a collection of environmental conditions (settings for the noise factors) is determined. This collection is called the *outer array*. Each run in the control factor design (*inner array*) is replicated within each of these environments. The mean and variance of the process over the outer array are computed for each run in the inner array. Either the outer array or the inner array may consist of all possible different settings for the associated factors, or they may be fractions of all possible settings.

You can replicate designs in this way by using data set names for the POINTREP= and DESIGNREP= options in the OUTPUT statement. If you construct a design for your control factors and you want to run a noise factor design for each run in the control factor design, specify the data set that holds the noise factor design (that is, the *outer array*) with the POINTREP= option in the OUTPUT statement. See Example 22.14 on page 642 for an example.

Confounding Rules

Confounding rules give the values of factors in terms of the values of the *run-indexing factors* for a design. (See "Types of Factors" on page 648 for a discussion of run-indexing factors.) The FACTEX procedure uses these rules to construct designs. The confounding rules also determine the alias structure of the design. To display the confounding rules for a design, use the CONFOUNDING option in the EXAMINE statement.

For two-level factors, the rules are displayed in a multiplicative notation using the default values of -1 and $+1$ for the factors. For example, the confounding rule

```
X8 = X1*X2*X3*X4*X5*X6*X7
```

means that the level of factor X8 is derived as the product of the levels of factors X1 through X7 for each run in the design. X8 will always have a value of +1 or -1 since these are the values of X1 through X7. For factors with $q > 2$ levels, confounding rules are printed in an additive notation, and the arithmetic is performed in the Galois field of size q. For example, in a design for three-level factors, the confounding rule

```
F = B + (2*C) + D + (2*E)
```

means that the level of factor F is computed by adding the levels of B and D and two times the levels of C and E, all modulo 3. Note that if q is not a prime number, Galois field arithmetic is not equivalent to arithmetic modulo q.

Blocks are introduced into designs by using *block pseudo-factors*. The confounding rule for the ith block pseudo-factor has `[Bi]` on the left-hand side.

For details on how confounding rules are constructed, see "Suitable Confounding Rules" on page 664.

Alias Structure

The alias structure of a design identifies which effects are confounded (or aliased) with each other in the design. Note the difference between alias structure and confounding rules: the confounding rules are used to construct the design, and the alias structure is a result of using a given set of confounding rules. To display the alias structure for a design, use the ALIAS option in the EXAMINE statement.

Examining the alias structure is important since aliased effects cannot be estimated separately from one another. When several effects are listed as equal, the effects are all jointly aliased with one another and form an *alias chain* or *alias string*. For example,

```
TEMP*MOIST = HPRESS*GATE = THICK*SCREW = BPRESS*TIME
```

is an alias chain that shows the relationship between four two-factor interactions. If you want separate estimates of TEMP*MOIST and THICK*SCREW, for instance, a

design with this alias chain would not be acceptable. Designs of even resolution $2k$ contain one or more such chains of confounded k-factor interactions.

By default, the FACTEX procedure displays alias chains with effects up to a certain order d, where main effects are order 1, two-factor interactions are order 2, and so on. The value of d can be specified in the ALIAS option , or you can use the default calculated by the procedure ; see page 608 for details. Alias chains that are confounded with blocks are displayed with [B] on the left-hand side.

Minimum Aberration

As discussed in "Speeding Up the Search" on page 667, the FACTEX procedure uses a tree search algorithm to find the confounding rules of a design that matches the size and resolution you specify. There may be more than one solution set of confounding rules, and usually the FACTEX procedure chooses the first one it finds. However, there can still be important differences between designs with the same resolution; to deal with these differences, Fries and Hunter (1980) introduced the concept of *aberration* in confounded fractional factorial designs. This section defines aberration and discusses how to request minimum aberration designs with the FACTEX procedure.

Recall that a design has resolution r if r is the smallest order of the interactions that are confounded with zero. The idea behind minimum aberration is that a resolution r design that confounds *as few* rth-order interactions as possible is preferable. Technically, the aberration of a design is the vector $\mathbf{k} = \{k_1, k_2, \ldots\}$, where k_i is the number of ith-order interactions that are confounded with zero. A design with aberration $\mathbf{k}$ has *minimum aberration* if $\mathbf{k} \leq \mathbf{k}'$ for any other design with aberration $\mathbf{k}'$, in the sense that $k_i < k'_i$ for the first i for which $k_i \neq k'_i$.

For example, consider the resolution 4 design for seven two-level factors in 32 runs (2^{7-2}_{IV}) discussed in Example 22.11 on page 633.

By specifying 5 for the order d for the ALIASING option, you can see how many fourth- and fifth-order interactions are confounded with zero. The default design constructed by the FACTEX procedure confounds two fourth-order interactions and no fifth-order interactions with zero.

```
0 = A*B*F*G = C*D*E*G
```

Thus, part of the aberration for this design is

$$\{k_3, k_4, k_5, \ldots\} \quad = \quad \{0, 2, 0, \ldots\}$$

On the other hand, the design constructed using the MINABS option confounds only one fourth-order interaction and two fifth-order interactions with zero.

```
0 = C*D*E*F = A*B*C*F*G = A*B*D*E*G
```

Thus, part of the aberration for this design is

$$\{k'_3, k'_4, k'_5, \ldots\} \quad = \quad \{0, 1, 2, \ldots\}$$

Since the two aberrations first differ for k_4 and k_4', and since $k_4' < k_4$, the aberration for the second design is less than the aberration for the first design.

The definition of aberration requires evaluating the number of ith-order interactions that are confounded with zero for all $i \leq k$, where k is the number of factors. Since there are q^k generalized interactions between k q-level factors, this evaluation can be prohibitive if there are many factors. Moreover, it is unnecessary if, as is usually the case, you are interested only in small-order interactions. Therefore, when you specify the MINABS option, by default the FACTEX procedure evaluates the aberration only up to order d, where d is the same as the default maximum order for listing the aliasing (see the specifications for the EXAMINE statement on page 608). You can set d to any level by specifying (d) immediately after the MINABS option; see page 611 for details.

The discussion so far has dealt only with fractional unblocked designs, but one more point to consider is the definition of aberration for block designs. Define a vector $\mathbf{b} = b_1, b_2, \ldots$ similar to the aberration vector $\mathbf{k}$, except that b_i is the number of ith-order interactions that are confounded with blocks. A block design with $\mathbf{k}$ and $\mathbf{b}$ has minimum aberration if

- $\mathbf{k}$ is minimum
- among all designs with minimum $\mathbf{k}$, $\mathbf{b}$ is minimum

Output

By default, the FACTEX procedure does not display any output. For each design that it constructs, the procedure displays a message in the SAS log that provides

- the number of runs in the design
- the number of blocks and the block size, if appropriate
- the maximum resolution of the design

If you use the DESIGN option in an EXAMINE statement, the procedure displays the coded runs in the design using standard values, as described in the "OUTPUT Statement" section on page 611. If you use the CONFOUNDING option in an EXAMINE statement, the procedure displays the confounding rules used to construct the design. If you use the ALIAS option in an EXAMINE statement, the procedure displays the alias structure for the design.

The FACTEX procedure also creates output data sets with the OUTPUT statement. Since the procedure is interactive, you can use many OUTPUT statements in a given run of the FACTEX procedure to produce many output data sets if you separate them with **`run;`** statements.

ODS Tables

The following table summarizes the ODS tables that you can request with the PROC FACTEX statement.

Table 22.8. ODS Tables Produced in PROC FACTEX

ODS Table Name	Description	Statement	Option
DesignPoints	Design points	EXAMINE	DESIGN
FactorRules	Treatment factor confounding rules	EXAMINE	CONFOUNDING
BlockRules	Block factor confounding rules	EXAMINE	CONFOUNDING
Aliasing	Alias structure	EXAMINE	ALIASING

Chapter 23
Theory of Orthogonal Designs

Chapter Contents

OVERVIEW 663

STRUCTURE OF GENERAL FACTORIAL DESIGNS 663

SUITABLE CONFOUNDING RULES 664
- Design Factors 664
- Block Factors 665
- General Criteria 666

SEARCHING FOR CONFOUNDING RULES 666

SPEEDING UP THE SEARCH 667

GENERAL RECOMMENDATIONS 668

Chapter 23
Theory of Orthogonal Designs

Overview

This chapter provides the mathematical and statistical background for designs constructed by the FACTEX procedure; it also outlines the search algorithm that is used to find suitable construction rules. Note that the material in this chapter is general and theoretical; you do not need to read this chapter to use the procedure for constructing most common experimental designs. On the other hand, you should read this chapter

- to understand the general structure of designs that can be constructed with the FACTEX procedure
- to construct designs for factors with more than two levels, especially if interactions are involved
- to improve the search used by the procedure when constructing complicated designs for many factors

Structure of General Factorial Designs

The FACTEX procedure constructs a fractional design for q-level factors using the *Galois field* (or *finite field*) of size q. This is a system with q elements and two operations $+$ and $\times$, which satisfy the usual mathematical axioms for addition and multiplication. When q is a prime number, finite field arithmetic is equivalent to regular integer arithmetic modulo q. When $q = 2$, addition of the two elements of the finite field is equivalent to multiplication of the integers $+1$ and -1. Since designs for factors with levels $+1$ and -1 are the factorial designs most commonly covered in textbooks, the arithmetic for fractional factorial designs is usually shown in multiplicative form. However, throughout this section a more general notation is used.

A design for q-level factors in q^m runs constructed by the FACTEX procedure has the following general form. The first m factors are taken to index the runs in the design, with one run for each different combination of the levels of these factors, where the levels run from 0 to $q - 1$. These factors are called *run-indexing factors*. For a particular run, the value F of any other factor in the design is derived from the levels $P_1, P_2, \ldots, P_m$ of the run-indexing factors by means of *confounding rules*. These rules are of the general form

$$F = r_1 P_1 + r_2 P_2 + \ldots + r_m P_m$$

where all the arithmetic is performed in the finite field of size q.

The linear combination on the right-hand side of the preceding equation is called a *generalized interaction* between the run-indexing factors. A generalized interaction is part of the statistical interaction between the factors with nonzero coefficients in the linear combination. The factor F is said to be *confounded* or *aliased* with this generalized interaction; two terms are confounded when the levels they take in the design yield identical partitions of the runs, so that their effects cannot be distinguished. The confounding rules characterize the design, and the problem of constructing the design reduces to finding suitable confounding rules.

Suitable Confounding Rules

Design Factors

This section explains how the criteria for a design can be reduced to prescribing that certain generalized interactions are *not* to be "confounded with zero."

Suitable confounding rules depend on the effects you want to estimate with the design. For example, if you want to estimate the main effects of both A and B, the following rule is inappropriate:

$$A \quad = \quad B$$

With this rule, the levels of A and B are the same in every run of the design, and the main effects of the two factors cannot be estimated independently of one another. Thus, the first criterion for a suitable confounding rule is that no two effects you want to estimate should be confounded with each other.

Furthermore, an effect you want to estimate should not be confounded with an effect that is nonnegligible. For example, if the interaction between C and D is nonnegligible and you want to estimate the main effect of A, the following confounding rule is inappropriate:

$$A \quad = \quad C + D$$

(Recall that this section uses a general linear form for confounding rules instead of the usual multiplicative form. For factors with levels $+1$ and -1, the preceding rule is equivalent to $A = C * D$.)

Another kind of confounding involves *confounding with zero*. If a factor or a generalized interaction F has the same value in every run of the design, then F is *confounded with zero*. Such confounding is denoted as

$$0 \quad = \quad F$$

Interactions are estimable with the design if and only if they are not confounded with zero. Consequently, another criterion for a suitable confounding rule is that no effect that you want to estimate can be confounded with zero. The confounding rule for two main effects

$$A \quad = \quad B$$

can be written as a generalized interaction confounded with zero.

$$0 \quad = \quad -A + B$$

The right-hand side of the preceding equation is part of the interaction between A and B. Thus, for any two effects to be unconfounded, it is equivalent to prescribe that no part of their generalized interaction be confounded with zero.

Note that it is not enough to make sure that only the confounding rules themselves satisfy these restrictions. The consequences of the confounding rules must also satisfy the restrictions. For example, suppose you want to make sure that main effects are not confounded with two-factor interactions, and suppose that the confounding rule for factor E is

$$E \quad = \quad A + B + C + D$$

Then the following rule cannot be used for factor F:

$$F \quad = \quad A + B + C$$

Even though the rule for F does not confound F with a two-factor interaction, this rule forces a generalized interaction between E and F to be aliased with the main effect of D, since

$$E - F \quad = \quad (A + B + C + D) - (A + B + C) \; = \; D$$

Block Factors

If your design involves blocks, additional confounding criteria need to be considered. Blocks are introduced into designs by means of *block pseudo-factors*. (See "Types of Factors" on page 648 for details.) A design for q-level factors in q^s blocks contains s block pseudo-factors. Denoting the levels of these factors for any given run by $B_1, B_2, \ldots, B_s$, the index of the block in which the run occurs is given by

$$B_1 + qB_2 + q^2B_3 + \ldots + q^{s-1}B_s$$

For each block to occur in the design, every possible combination of block pseudo-factors must occur. This can happen only if all main effects and interactions between the block factors are estimable, which leads to yet another criterion for the confounding rules. Moreover, the effects you want to estimate cannot be confounded with blocks. In general,

- no generalized block pseudo-factors can be confounded with zero
- no generalized interactions between block pseudo-factors and effects you want to estimate can be confounded with zero

General Criteria

The criteria for an orthogonally confounded q^k design reduce to requiring that no generalized interactions in a certain set $\mathcal{M}$ can be confounded with zero. (See "Structure of General Factorial Designs" on page 663 for a definition of *generalized interaction*.) This section presents the general definition of $\mathcal{M}$. First, define three sets, as follows:

$\mathcal{E}$	the set of effects that you want to estimate
$\mathcal{N}$	the set of effects you do not want to estimate but that have unknown nonzero magnitudes (referred to as *nonnegligible* effects)
$\mathcal{B}$	the set of all generalized interactions between block pseudo-factors

Furthermore, for any two sets of effects $\mathcal{A}$ and $\mathcal{B}$, denote by $\mathcal{A} \times \mathcal{B}$ the set of all generalized interactions between the effects in $\mathcal{A}$ and the effects in $\mathcal{B}$.

Then the general rules for creating the set of effects $\mathcal{M}$ that are not to be confounded with zero are as follows:

- Put $\mathcal{E}$ in $\mathcal{M}$. This ensures that all effects in $\mathcal{E}$ are estimable.
- Put $\mathcal{E} \times \mathcal{E}$ in $\mathcal{M}$. This ensures that all pairs of effects in $\mathcal{E}$ are unconfounded with each other.
- Put $\mathcal{E} \times \mathcal{N}$ in $\mathcal{M}$. This ensures that effects in $\mathcal{E}$ are unconfounded with effects in $\mathcal{N}$.
- Put $\mathcal{B}$ in $\mathcal{M}$. This ensures that all q^s blocks occur in the design.
- Put $\mathcal{E} \times \mathcal{B}$ in $\mathcal{M}$. This ensures that effects in $\mathcal{E}$ are unconfounded with blocks.

Searching for Confounding Rules

The goal in constructing a design, then, is to find confounding rules that do not confound with zero any of the effects in the set $\mathcal{M}$ defined previously. This section describes the sequential search performed by the FACTEX procedure to accomplish this goal.

First, construct the set C_1 of candidates for the first confounding rule, taking into account the set $\mathcal{M}$ of effects not to be confounded with zero. If C_1 is empty, then no design is possible; otherwise, choose one of the candidates $r_1 \in C_1$ for the first confounding rule and construct the set C_2 of candidates for the second confounding rule, taking both $\mathcal{M}$ and r_1 into account. If C_2 is empty, choose another candidate from C_1; otherwise, choose one of the candidates rules $r_2 \in C_2$ and go on to the third rule. The search continues either until it succeeds in finding a rule for every non-run-indexing factor or the search fails because the set C_1 is exhausted.

The algorithm used by the FACTEX procedure to select confounding rules is essentially a depth-first tree search. Imagine a tree structure in which the branches

connected to the root node correspond to the candidates C_1. Traversing one of these branches corresponds to choosing the corresponding rule r_1 from C_1. The branches attached to the node at the next level correspond to the candidates for the second rule given r_1. In general, each node at level i of the tree corresponds to a set of feasible choices for rules $r_1, \ldots, r_i$, and the rest of the tree above this node corresponds to the set of all possible feasible choices for the rest of the rules.

Speeding Up the Search

For designs with many factors or blocks, the tree of candidate confounding rules can be very large and the search can take a very long time. In these cases, the FACTEX procedure spends a lot of time exploring sets of rules that are essentially the same and that all result in failure. A technique for pruning the search tree (see Figure 23.1) is as follows. Suppose that for some selection r_i for rule i, all the branches for the next rule eventually result in failure. Then any other selection r_i' is immediately declared a failure if the resulting number of candidates is the same as for the failed rule r_i. The search goes on to the next selection for rule i.

This method of pruning is not perfect; it may prune a branch of the search tree that would have resulted in a success. In mathematical terms, candidate sets C_i are not necessarily isomorphic because they have the same size. You can use the NOCHECK option in the PROC FACTEX statement to turn off the pruning. With the NOCHECK option, the FACTEX procedure searches the entire tree of feasible confounding rules; and if given enough time, will find a design if one exists. The default argument for the TIME= option on the PROC FACTEX statement limits the search time to one minute.

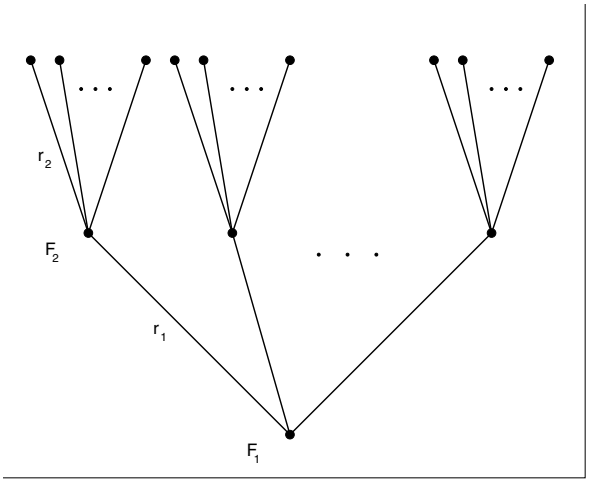


Figure 23.1. Search Tree

On the other hand, you should recognize how rarely the NOCHECK option is needed to produce a design with a given resolution. For example, consider all possible blocked and unblocked two-level designs with minimum resolution for 50 or fewer factors and 128 or fewer runs. Of the 849 different designs, the NOCHECK option is required in only five cases. The five designs for which the NOCHECK option is

required are listed in Table 23.1. Note that all of these are block designs, most for many factors and relatively small blocks.

Table 23.1. Designs Requiring the NOCHECK Option

Number of Factors	Number of Runs	Block Size	Resolution
5	16	2	4
21	32	4	3
22	32	4	3
23	32	4	3
39	64	4	3

General Recommendations

Choosing appropriate confounding rules can be difficult, especially if the set $\mathcal{M}$ is at all complicated. Even if a design is found that satisfies the model specification, it is a good idea to examine the alias structure to make sure that you understand the alias structure generated by the confounding rules. To do so, use the ALIAS option in the EXAMINE statement.

For more details on the general mathematical theory of orthogonal factorial designs, refer to Bose (1947).

References

Addelman, S. (1962), "Orthogonal Main-Effects Plans for Asymmetrical Factorial Experiments," *Technometrics*, 4, 21–46.

Bose, R.C. (1947), "Mathematical Theory of the Symmetrical Factorial Design," *Sankhya*, 8, 107–166.

Box, G.E.P. and Bisgaard, S. (1987), "The Scientific Context of Quality Improvement," *Quality Progress*, 20(6), 54–61.

Box, G.E.P., Hunter, W.G., and Hunter, J.S. (1978), *Statistics for Experimenters*, New York: John Wiley & Sons, Inc.

Byrne, D.M. and Taguchi, S. (1986), "The Taguchi Approach to Parameter Design," *Quality Congress Transactions*, American Society for Quality Control, 177, 168–177.

Chakravarti, I.M. (1956), "Fractional Replication in Asymmetrical Factorial Designs and Partially Balanced Arrays," *Sankhya*, 17,143–164.

Cochran, W.G. and Cox, G.M. (1957), *Experimental Designs, Second Edition*, New York: John Wiley & Sons, Inc.

Dehnad, K. (1989), *Quality Control, Robust Design, and the Taguchi Methods*, Pacific Grove, California: Wadsworth and Brooks.

Fries, A. and Hunter, W.G. (1980), "Minimum Aberration 2^{k-p} Designs," *Technometrics*, 22 (4), 601–608.

Hogg, R.V. and Ledolter J. (1992), *Applied Statistics for Engineers and Physical Scientists, Second Edition,* New York: Macmillan Publishing Company, Inc.

Hunter, J.S. (1985), "Statistical Design Applied to Product Design," *Journal of Quality Technology*, 17, 210–221.

John, P.W.M. (1972), *Statistical Design and Analysis of Experiments*, New York: Macmillan Publishing Company, Inc.

Kackar, R. (1985), "Off-line Quality Control, Parameter Design, and the Taguchi Method," *Journal of Quality Technology*, 17, 176–209.

Kempthorne, O. (1975), *The Design and Analysis of Experiments*, Huntington, NY: Robert E. Krieger Publishing Co.

Margolin, B.H. (1967), "Systematic Methods of Analyzing $2^n \times 3^m$ Factorial Experiments with Applications," *Technometrics*, 11, 431–444.

Mason, R.L., Gunst, R.F., and Hess J.L. (1989), *Statistical Design and Analysis of Experiments*, New York: John Wiley and Sons, Inc.

Montgomery, D.C. (1991), *Design and Analysis of Experiments, Third Edition,* New York: John Wiley & Sons, Inc.

Phadke, M. (1989), *Quality Engineering Using Robust Design*, Englewood Cliffs, New Jersey: Prentice Hall.

SAS Institute Inc. (1999), *SAS/STAT User's Guide, Version 8*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1999), *SAS Language Reference: Dictionary, Version 8*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1999), *Getting Started with the ADX Interface for Design of Experiments*, Cary, NC: SAS Institute Inc.

Searle, S.R. (1971), *Linear Models*, New York: John Wiley & Sons, Inc.

Taguchi, G. and Wu, Y. (1980), *Introduction to Off-line Quality Control*, Nagoya, Japan: Central Japan Quality Control Association.

Williams, E.J. (1949), "Experimental Designs Balanced for the Estimation of Residual Effects of Treatments," *Australian Journal of Scientific Research*, Series A, 2, 149–168.

Yates, F. (1936), "Incomplete Randomized Blocks," *Annals of Eugenics*, 7, 121–140.

Yin, G.Z. and Jillie, D.W. (1987), "Orthogonal Design for Process Optimization and its Application in Plasma Etching," *Solid State Technology*, May, 127–132.

Part 5
The ISHIKAWA Procedure

Contents

Chapter 24. Introduction 673

Chapter 25. Details of the ISHIKAWA Environment 687

References 749

Chapter 24
Introduction

Chapter Contents

OVERVIEW 675

TERMINOLOGY 677

TUTORIAL 679

Chapter 24
Introduction to the ISHIKAWA Environment

Overview

The Ishikawa diagram,* also known as a cause-and-effect diagram or fishbone diagram, is one of the seven basic tools for quality improvement in Japanese industry. It is used to display the factors that affect a particular quality characteristic or problem. For example, the following Ishikawa diagram shows factors affecting the quality of air travel service:

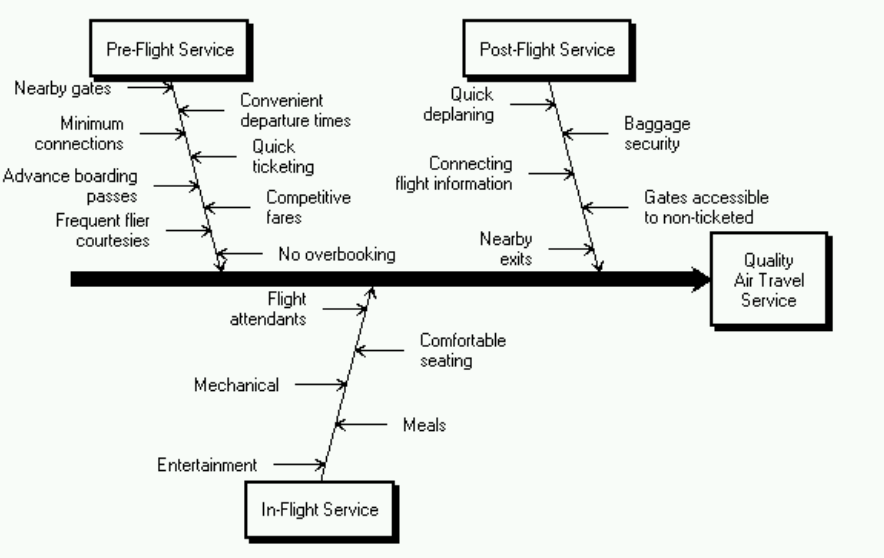


Figure 24.1. Ishikawa Diagram

In this example, the factors are organized into three categories of service (Pre-flight, In-flight, and Post-flight), which are represented as branches. The factors affecting each of these areas are represented as stems.

An Ishikawa diagram is typically the result of a brainstorming session to improve a product, process, or service. The main goal is represented by a main arrow or trunk, and primary factors are represented as sub-arrows or branches. Secondary factors are then added as stems, tertiary factors as leaves, and so on.

Creating the diagram stimulates discussion and often leads to an increased understanding of a complex problem. Japanese QC Circle members often post Ishikawa diagrams in a display area where they will be accessible to managers and other groups;

* The Ishikawa diagram is named after its developer, Kaoru Ishikawa (1915-1989), a leader in Japanese quality control; refer to Karabatsos (1989), Kume (1985) and Sarazen (1990).

refer to Rodriguez (1991). In the United States, Ishikawa diagrams are often included in presentations by plant personnel to management or customers.

Traditionally, Ishikawa diagrams have been prepared by hand on paper or chalk boards. This limits the amount of detail that can be added and makes it awkward to update the diagram as an understanding of the process evolves. Manual preparation also restricts the collection and display of data on the diagram, as advocated by Ishikawa (1982).

The ISHIKAWA procedure was designed to overcome these limitations by providing a highly interactive graphics environment (referred to in this section as the *ISHIKAWA environment*) for creating and modifying Ishikawa diagrams.

In the ISHIKAWA environment you can

- add and delete arrows with a mouse. You can also swap, copy, and so forth.
- highlight special problems or critical paths with line styles and color
- display additional data for each of the arrows in a popup notepad
- display portions of the diagram in separate windows for increasing or isolating detail. You can also divide sections of the diagram into separate Ishikawa diagrams.
- merge multiple Ishikawa diagrams into a single, master diagram
- display any number of arrows and up to ten levels of detail
- foliate and defoliate diagrams dynamically
- save diagrams for future editing
- save diagrams in graphics catalogs or export them to host clipboards or graphics files
- customize graphical features such as fonts, arrow types, and box styles
- obtain online help at any time

If you are using the ISHIKAWA procedure for the first time, the tutorial, at the end of this chapter, demonstrates some of the basic operations used in the ISHIKAWA procedure. A summary of these operations (and others) can be found in the section "Summary of Operations" on page 689.

For a detailed discussion of each of the operations, see Chapter 25, "Details of the ISHIKAWA Environment," starting on page 687. This chapter includes many tools not presented in the tutorial.

Terminology

This section introduces basic operations used in the ISHIKAWA environment and defines terms used to describe the ISHIKAWA procedure. Some details depend on your *host*, which is the specific system of computing hardware and software you use. For example, all hosts present the ISHIKAWA environment in a system of *windows* on the host's *display*, but the appearance of your windows may differ from the figures in this book. You can find more information in the SAS companion for your host and in your host system documentation.

Using a Mouse

On most hosts you can use a *mouse* to point to objects on the display. A mouse is a physical device that controls the location of a *cursor*, which is a small, movable symbol on the display. Due to the precision required, you must use a mouse to perform tasks in the ISHIKAWA environment.

Text is placed relative to the *text* cursor (▮) and not the *mouse* cursor (↖). The mouse cursor is always visible, while the text cursor is displayed only when text can be entered (for example, when an arrow is being added).

The mouse also has *buttons* that work like keys on the keyboard. On most hosts, you *select* an object by pointing to it with the mouse and clicking the left button on the mouse. To *click*, press the button down and release it quickly without moving the mouse. To *double click*, click *twice* quickly without moving the mouse. To *drag*, move the mouse while holding down the left mouse button.

Popup menus appear to *pop up* on the display when you press a button—usually the right mouse button. Popup menus are convenient to use, since they always appear at the cursor location. Selecting an item from the popup menu, however, is host specific.

For details about using the mouse on your system, consult the SAS companion for your host.

Using Context-Sensitive Operations

Basic operations such as add, edit, delete, and move are invoked by activating the mouse near various *hotspots* along the arrows rather than selecting tools from a tools palette. The hotspots are the following context-sensitive areas in the Ishikawa diagram:

- arrow heads, tails, lines, and labels
- window background

Given such evident features, and a rigidly defined structure, the hotspots are not highlighted.

The hotspot areas are illustrated in the following figure:

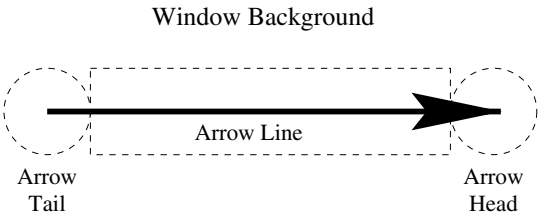


Figure 24.2. Context-Sensitive Locations (Hotspots)

The dotted, circular region at the right end of the arrow is the arrow head hotspot. Arrows attach to other arrows at the head of the arrow. The dotted, circular region at the left end of the arrow is the arrow tail hotspot. The region that encompasses the arrow line is also hot. Every arrow in the diagram has these hotspots.

The window background is any area inside the window and outside the dotted lines. You use the window background to cancel pending operations (such as adds and moves) and to control global or environment-specific operations (such as decreasing detail and tagging arrows).

When you activate the mouse, the ISHIKAWA environment uses the mouse event (click, double click, drag, or popup) and the hotspot type (head, tail, line, label, or background) to infer the intended operation. The ISHIKAWA environment responds differently depending upon which hotspot you select and how you select it. This is often referred to as *context-sensitive* behavior.

Context sensitivity allows the ISHIKAWA environment to operate without modes. In a modeless environment like the ISHIKAWA environment, context-sensitive operations reduce the amount of mouse travel (the time and distance spent moving the cursor from the drawing area to the tools palette and back). For example, you do not go to a tools palette to change from *add mode* to *delete mode*. This allows you to focus on the diagram rather than on the diagramming tool.

In the ISHIKAWA environment, the primary operations such as add, edit, delete, and move are all operations associated with a specific hotspot and the mouse button. Secondary operations such as zoom, copy, highlight, and so forth operate from *context-sensitive* popup menus (typically activated using the right mouse button.) Other, less frequently used operations are available from the command bar.

The relationship between these context-sensitive areas, the mouse actions, and the basic ISHIKAWA tools is introduced in the tutorial that follows. A comprehensive discussion of each operation is given in Chapter 25, "Details of the ISHIKAWA Environment," starting on page 687. In addition, the tables beginning on page 689 provide a good overview of how to function inside the ISHIKAWA environment.

Using the Command Bar

In addition to the editing tools, the ISHIKAWA environment provides a number of file management, printing, and help facilities. These facilities are located on the *pull-down* menu associated with the window. The appearance and location of the command bar are host specific. On most hosts, you choose these operations by *pulling*

down a menu from a *menu bar* using the mouse button. For more details about using the command bar on your system, consult the SAS companion for your host.

Tutorial

The following example is used throughout the ISHIKAWA chapters. Later examples illustrate how to add to and modify this diagram. If you are not familiar with the ISHIKAWA procedure, you may want to complete this tutorial before proceeding to Chapter 25, "Details of the ISHIKAWA Environment," starting on page 687. In this tutorial you will learn to create and save a simple Ishikawa diagram.

See ISHPLANE in the SAS/QC Sample Library

A task force is studying ways to improve the quality of passenger service for a major airline. After a preliminary discussion, the team concludes that three major areas should be considered: pre-flight service, in-flight service, and post-flight service. This result is to be displayed with the following preliminary Ishikawa diagram:

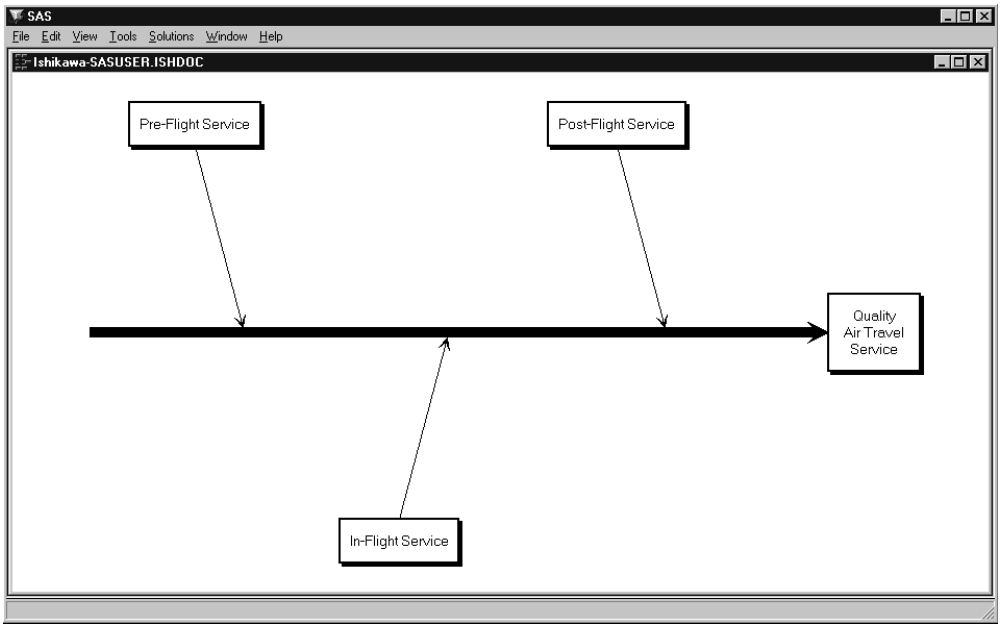


Figure 24.3. Preliminary Ishikawa Diagram

1. To begin using the ISHIKAWA environment, submit the following SAS statements:

```
proc ishikawa;
   run;
```

An initial menu appears on your display, as follows:

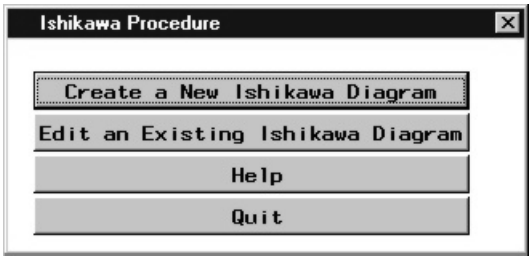


Figure 24.4. Initial Menu

2. Select **Create a New Ishikawa Diagram** to open a window containing a template for a new Ishikawa diagram.

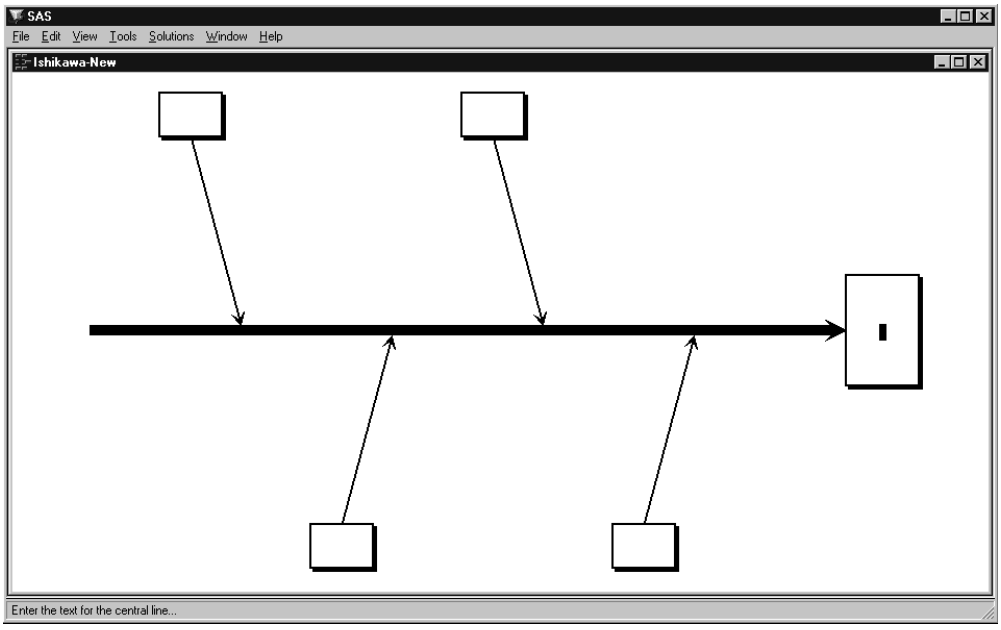


Figure 24.5. Starting a New Ishikawa Diagram

The ISHIKAWA environment guides you through the first steps of the diagramming process by prompting you to enter the text for the central line and then the upper left branch. During each step, a message indicating the action required is displayed in the message area for this window. Once you have completed these preliminary steps, you can proceed in any order you want.

3. Initially, the text cursor (▮) is positioned inside the box for the trunk. A message is displayed directing you to enter the first line of text for the trunk. Type the word *Quality*.*

*You can skip this step, in future diagrams, by pressing **Return** before entering any text.

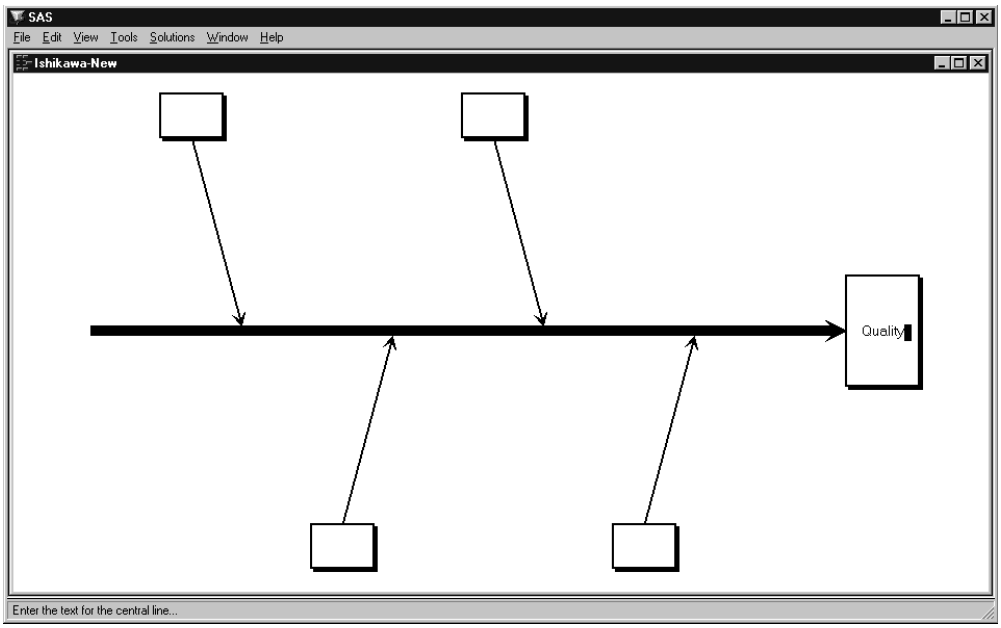


Figure 24.6. Labeling the Trunk

Note that the text is placed relative to the text cursor (▮). You can correct mistakes by using any of the keyboard editing keys or cursor navigation keys (for instance, **Back space** and ←).

4. Advance to the next line by pressing **Return** . Now complete the label by entering *Air Travel* and *Service* on separate lines.

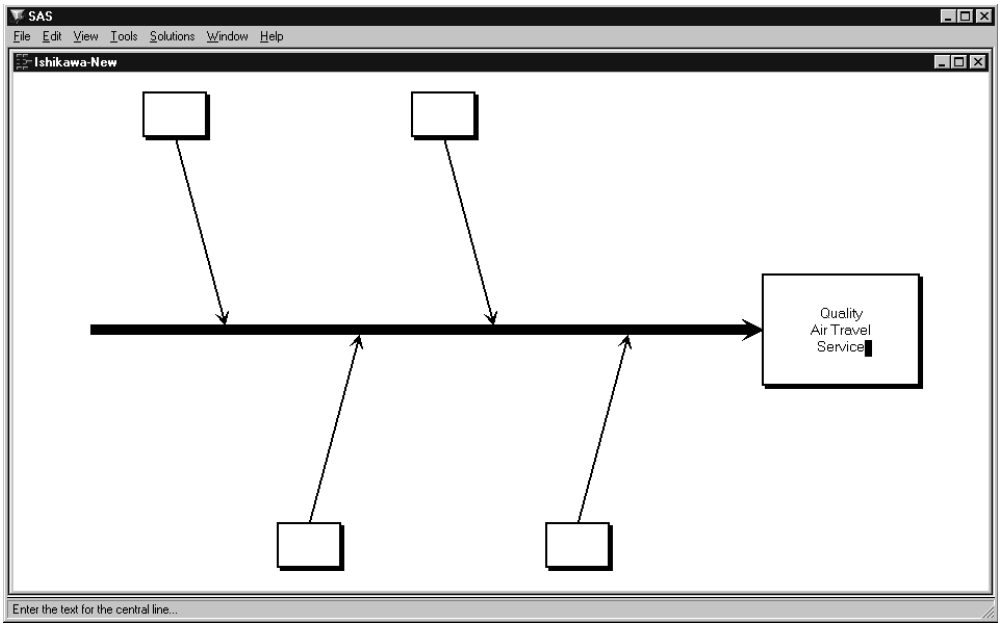


Figure 24.7. Labeling the Trunk *(continued)*

5. To terminate text entry, press **Return** a second time. The ISHIKAWA environment automatically moves the cursor to the upper left branch. If you made a mistake labeling the trunk, continue with the example. You cannot return to the trunk until you have finished the branch.

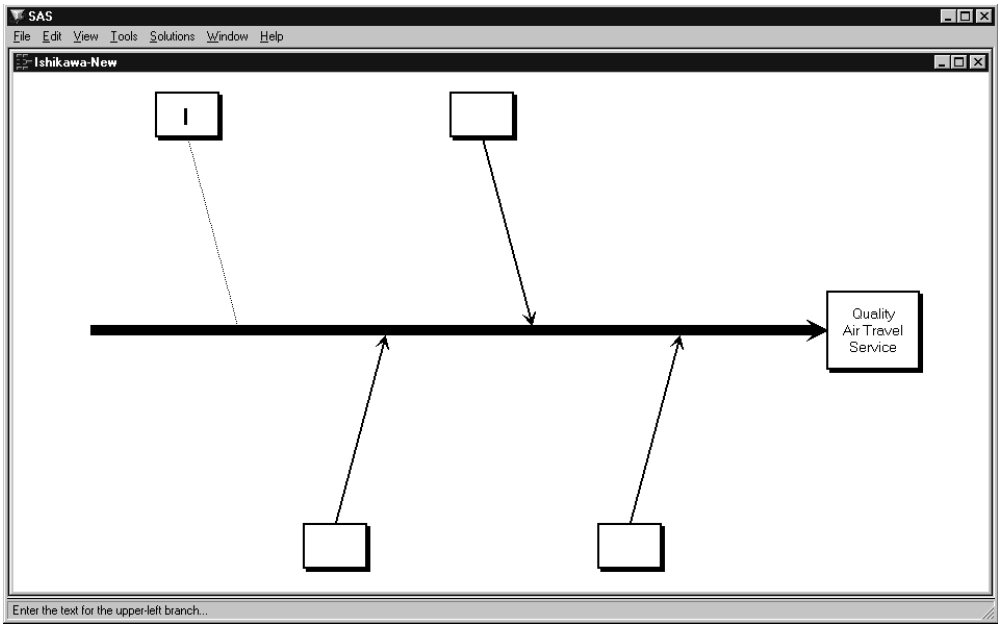


Figure 24.8. Labeling the First Branch

6. Enter the label *Pre-Flight Service*. Press **Return** twice to terminate text entry for this branch.* Your window should now look like this:

*You can skip this step, in future diagrams, by pressing **Return** before entering any text.

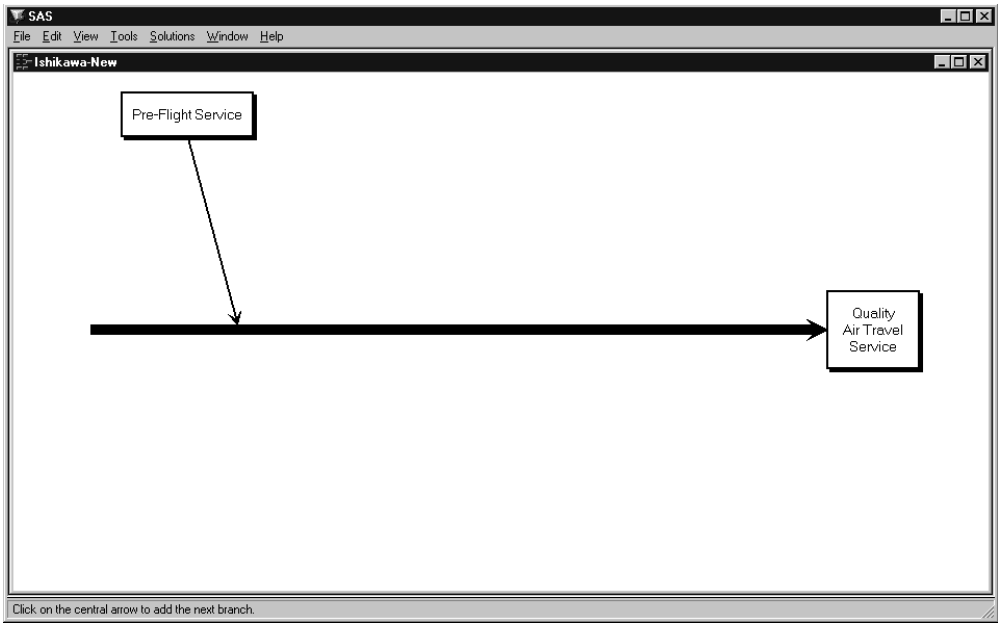


Figure 24.9. Completed Branch Label

Note that when you finish entering text for the upper left branch, the other branches are deleted. These were temporarily displayed as visual cues, and now it is up to you to decide where to add the remaining branches.

7. To add the branch labeled *In-Flight Service* to the lower half of the diagram, position the cursor slightly below the point where you want the branch to attach to the trunk and click the mouse button. The branch appears with the text cursor centered inside the box. Enter the first line of text.

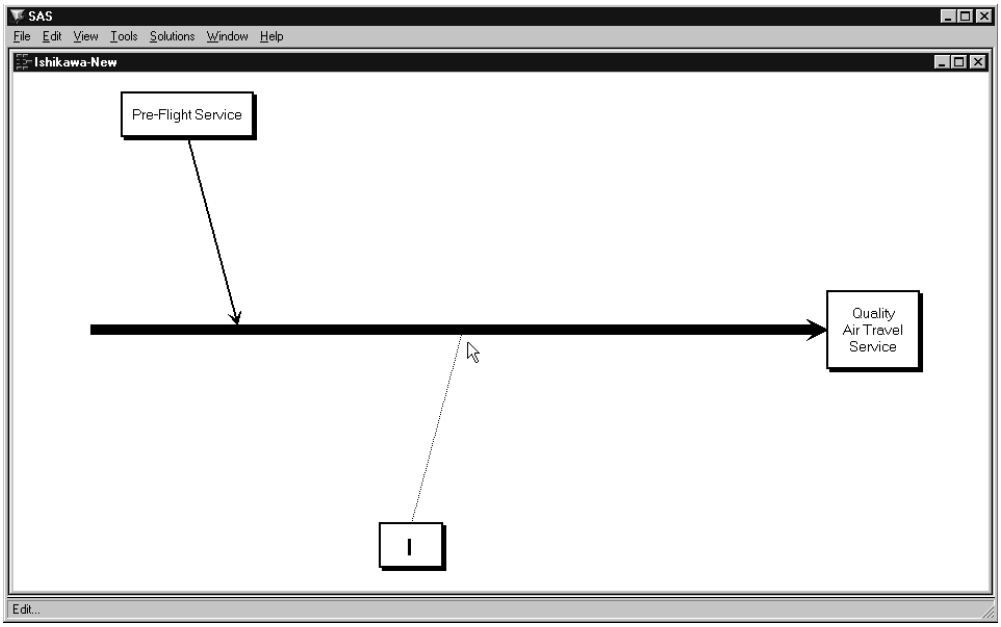


Figure 24.10. Adding a New Branch

If your branch is not positioned where you want it, move the cursor to the appropriate position along the trunk and click. Each time you click, the branch is moved to the new location.

If, on the other hand, the branch is not drawn at all, the cursor was probably too far away from the trunk to be recognized. Move the cursor closer to the trunk and try again.

Enter the label *In-Flight Service* and press **Return** twice to terminate text entry.

8. Next add the branch *Post-Flight Service* to the upper half of the diagram. Position the cursor so that it is just above the point where you want the branch to attach to the trunk and click.

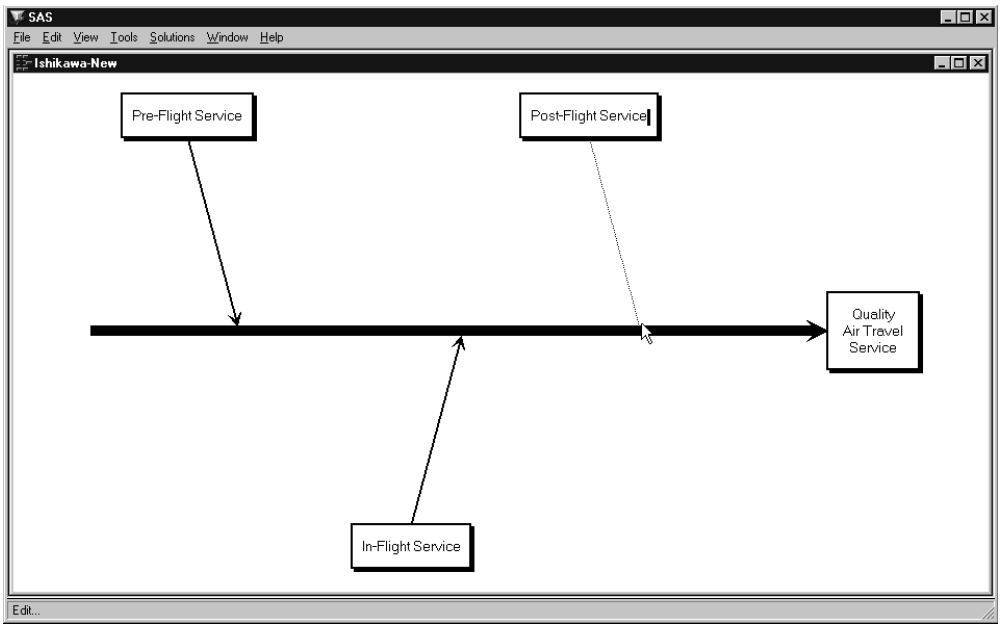


Figure 24.11. Adding the Last Branch

9. Press **Return** twice to terminate text entry.

Congratulations. You have just completed your first Ishikawa diagram using the ISHIKAWA procedure. In the process you learned to add branches to a diagram using context-sensitive mouse clicks. Future examples will illustrate other context sensitive areas, tools, and popup menus.

The examples that follow will, for the most part, expand on this diagram. To save the diagram, select **File** ▷ | **Save as** ▷ | **Data Set** from the command bar. Use SASUSER for the library name and AIRLINE for the data set name. Then select **Save**.

To leave the ISHIKAWA environment and return to the SAS Display Manager, select **File** ▷ | **Close** from the command bar.

Chapter 25
Details of the ISHIKAWA Environment

Chapter Contents

SUMMARY OF OPERATIONS 689

OPERATIONS 691
- Adding Arrows 691
- Labeling Arrows 694
- Moving Arrows 697
- Deleting Arrows 702
- Resizing Arrows 704
- Swapping Arrows 707
- Balancing Arrows 709
- Notepads 715
- Managing Complexity 716
- Zooming Arrows 719
- Isolating Arrows 720
- Merging Diagrams 721
- Creating Graphics Output Using SAS/GRAPH Software 724
- Creating Bitmap Graphics Output 726
- Modifying Fonts 727
- Modifying Box Colors 728
- Modifying Arrow Colors and Line Styles 729
- Modifying Text Colors 736
- Modifying Arrow Heads 736
- Modifying Environmental Attributes 737
- Saving an Ishikawa Diagram for Future Editing 738
- Reading an Existing Ishikawa Diagram 739
- Displaying Multiple Ishikawa Diagrams 740

INPUT AND OUTPUT DATA SETS 742

SYNTAX 744

EXAMPLES 745
- Example 25.1. Quality of Air Travel Service 745
- Example 25.2. Integrated Circuit Failures 746
- Example 25.3. Photographic Development Process 747

Chapter 25
Details of the ISHIKAWA Environment

This chapter presents detailed information about and examples of all the operations available in the ISHIKAWA environment. Some of the examples build upon the diagram created in the tutorial.

Summary of Operations

To invoke the following context-sensitive operations, apply the specified action (mouse event) to the appropriate hotspot, using the left mouse button:

Table 25.1. Primary Operations

Operation	Mouse Event	Hotspot	Page
Add	Click	Near the intended attachment point	691
Edit	Click	Arrow tail	694
Move	Click *(to pick)*	Arrow head	
	Click *(to drop)*	Near the intended attachment point	697
Delete	Double click	Arrow head	702
Resize	Drag	Arrow tail	704
Notepad	Double click	Arrow tail	715

To invoke the following operations, make the specified selection from the appropriate context-sensitive popup menu using the right mouse button:

Table 25.2. Secondary Operations

Operation	Menu	Selection	Page
Swap	Head or tail	**Swap**	707
Balance	Head or tail	**Balance**	709
Hide Detail	Background	**< Detail**	716
Show Detail	Background	**> Detail**	716
Zoom	Head or tail	**Zoom**	719
Isolate	Head or tail	**Isolate**	720
Print	Pull-down	**File** ▷ **Save as** ▷ **Graph**	724
Save	Pull-down	**File** ▷ **Save as** ▷ **Data Set**	738
Save	Pull-down	**File** ▷ **Save as** ▷ **Image**	.
Subset	Head or tail	**Subset**	729
Copy	Head or tail	**Copy**	721
Refresh	Background	**Refresh**	.
Unsubset	Background	**Unsubset**	729
Unbalance	Background	**Unbalance**	709
Undelete	Background	**Undelete**	702

When applied to the appropriate hotspots, the following actions (mouse events) invoke these context-sensitive operations:

Table 25.3. Context-Sensitive Tools

Hotspot	Mouse Event	Operation	Page
Arrow Head	Click	Begin move	697
	Double click	Delete	702
	Drag	Resize	704
	Popup menu	Subset	729
		Balance	709
		Swap	707
		Copy	721
		Zoom	719
		Isolate	720
Arrow Tail	Click	Edit	694
	Double click	Notepad	715
	Drag	Resize	704
	Popup menu	Subset	729
		Balance	709
		Swap	707
		Copy	721
Arrow	Click	Add new arrow or	691
		complete move operation	697
Window Background	Click	Drop (finish) pending action	.
	Drag	Drop (finish) pending action	.
	Popup menu	Undelete	702
		Unsubset	729
		Unbalance	709
		Show Detail	716
		Hide Detail	716
		Refresh	.

The File menu on the command bar can be used to control the following operations:

Table 25.4. File Menu

File ▷		Description	Page
New		Start a new diagram	679
Open		Open an existing diagram	739
Close		Close the current window	.
Merge		Merge in an existing diagram	721
Save as ▷	Data Set	Save as a SAS data set	738
	Graph	Print using SAS/GRAPH software	724
	Image	Save as an IMAGE, catalog entry	726
Export as Bitmap ▷	File...	Copy to a bitmap file	726
	Customize...	Export options	727

The Edit menu on the command bar can be used to control the following operations:

Table 25.5. Edit Menu

Edit ▷	Description	Page
Copy	Copy the diagram to host clipboard	726
Clear...	Clear the window	.

The View menu on the command bar can be used to control the following operations:

Table 25.6. View Menu

View ▷		Description	Page
Ishikawa Settings ▷	Palettes	Line and color palettes	729
	Background Color	Change window background color	.
	Save Attributes	Save window attributes: size, fonts and background color	.
	Balance Method	Select a balancing style	709
	Resize Method	Select a resizing method	704
	Primary Fonts...	Font dialog for first 3 levels	727
	Secondary Fonts...	Font dialog for levels 4-10	727
	Colors...	Color dialog	728
	Arrows...	Arrow style dialog	736
	Other...	Style dialog	737
Refresh		Refresh the window	.

The Help menu on the command bar can be used to control the following operations:

Table 25.7. Help Menu

Help ▷	Description
SAS System Help	SAS help system
Using This Window	Ishikawa specific help

Operations

This section provides details concerning the operations available in the ISHIKAWA environment. The order in which the topics appear is the order in which the operations are typically encountered. Some of the examples in this section build upon the diagram created in the tutorial.

Adding Arrows

You add an arrow by pointing with the mouse to the intended attachment point along an existing arrow and clicking the mouse. You control the direction of the new arrow by offsetting the mouse cursor a small distance away from the parent arrow on the side where the new arrow is to appear.

For example, to add upper branches, you offset the cursor slightly above the trunk. To add lower branches, you offset the cursor slightly below the trunk. Likewise, you offset the cursor to the right of the branch to add a right-hand stem and slightly left for a left-hand stem.

If a new arrow is not drawn as you intended (either positionally or directionally), you can easily move or delete it. To delete a new arrow before you have entered any text, click in the background. To move a new arrow before you have entered any text, move the cursor to a new attachment point and click.

Once an arrow is drawn, you are immediately prompted for its label (note the hint, *Edit...*, displayed on the message line and the appearance of the text cursor at the end of the arrow). See "Labeling Arrows" on page 694, for details on the text editing features of the ISHIKAWA environment.

A diagram can contain up to ten levels of detail, but the number of arrows is limited only by the resolution and size of your graphics display.

Example

Continuing with the tutorial example from "Tutorial" on page 679, suppose that you have obtained detailed information for each of the three major service areas, which you want to display by adding stems to the branches of the diagram you previously created. If you closed the ISHIKAWA environment after saving the data set, SASUSER.AIRLINE, you can easily restore the diagram by submitting:

```
proc ishikawa data=sasuser.airline;
   run;
```

To add a stem to the left side of the branch labeled *In-Flight Service*, position the cursor so that it is just to the left of the point where you want the stem to attach. Click the mouse. The new arrow (pending text) appears as follows:

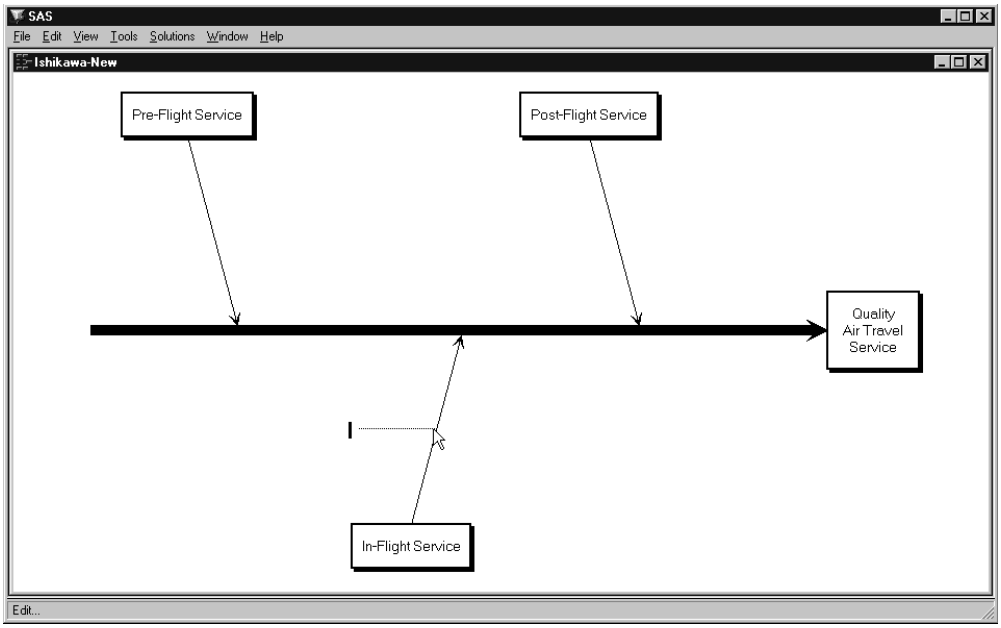


Figure 25.1. Adding the Left Stem

Type the label *Meals* and press **Return** twice.

To add a stem to the right side of the same branch, position the cursor so that it is just to the right of the attachment point. When you click the mouse, your window will appear as follows:

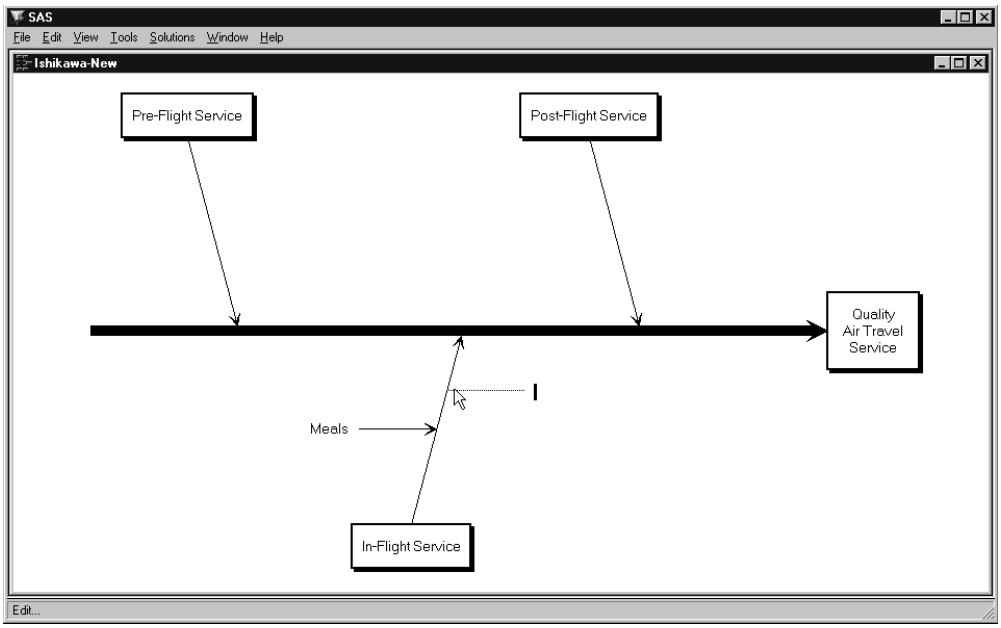


Figure 25.2. Adding the Right Stem

Type the label *Flight attendants* on two lines and press **Return** to terminate text entry.

Complete the diagram by adding the remaining stems shown in the following window:

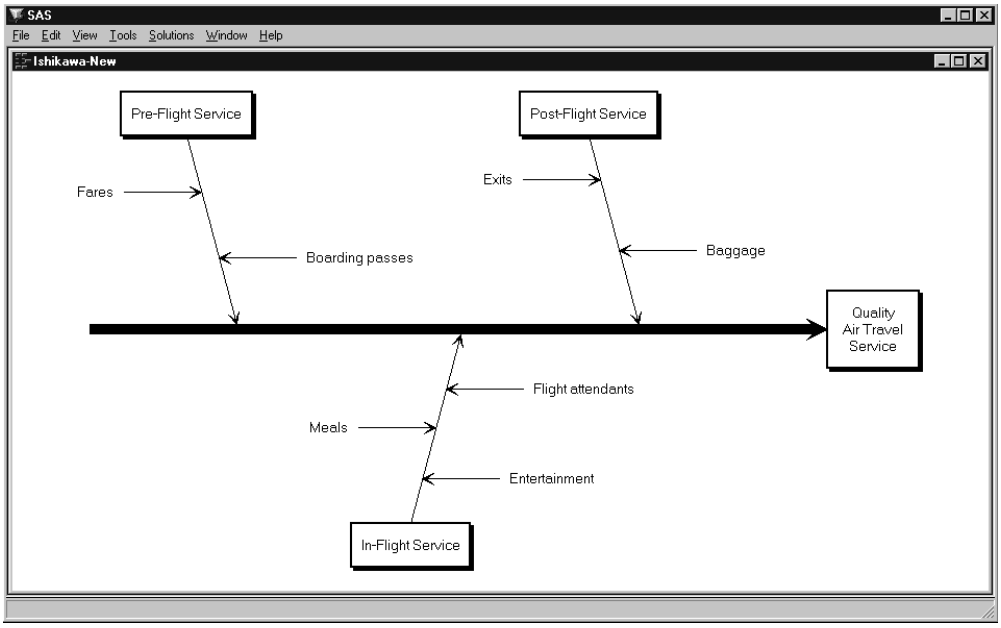


Figure 25.3. Stem-Level Diagram

Experiment further by adding several of the leaves shown in the following window. Don't be concerned if some of the labels collide with each other. Later, you will learn how to move and resize arrows.

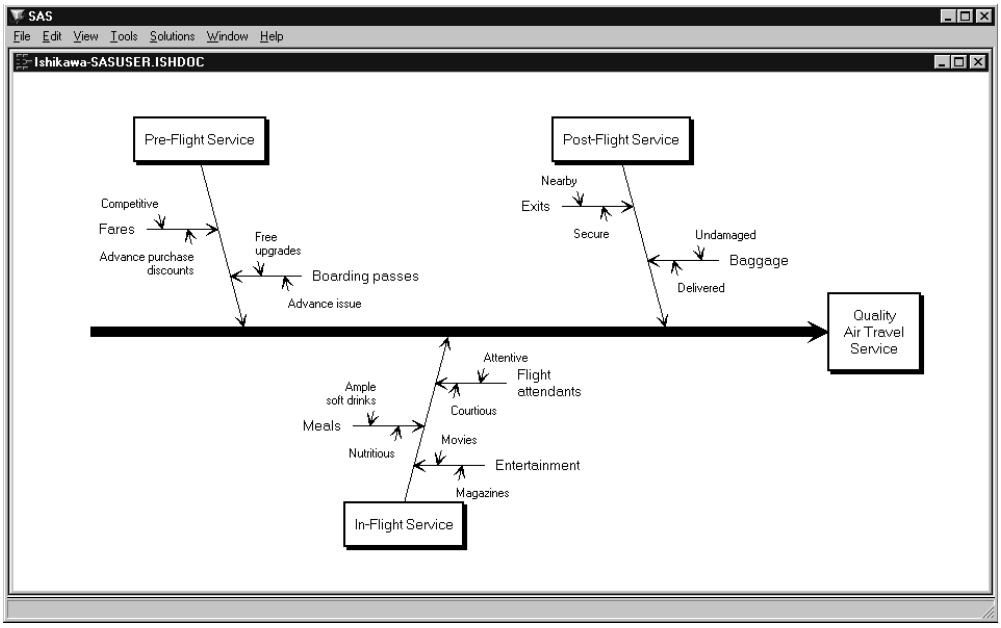


Figure 25.4. Leaf-Level Diagram

Labeling Arrows

To edit the label of an existing arrow, click on one of the following areas:

- the label
- the arrow tail (if the arrow does not have a label)
- inside the box for trunk and branch labels

Use your keyboard to enter the text.

On hosts that support direct graphical text entry,* the following functions are supported:

- edit keys such as **Back space**, **Delete char**, **Delete line**, and **Return**
- cursor navigation keys such as ↑, ↓, →, and ←
- the **Insert** key to toggle between insert and overstrike modes
- buffers to copy, cut, or paste text into and from external sources

*Devices such as the IBM3179 do not support the direct graphical text entry mechanism described in these examples. Instead, a text entry window pops up whenever you select an arrow for editing. You must edit the text for the arrow from the dialog box and close the text entry window before the diagram is updated.

Text entry is terminated whenever you press **Return** on an empty line or exceed the maximum line limit for a label. Text entry is also terminated whenever you click the mouse. This shifts focus away from the editing operation and to the new location.

Labels are restricted to 40 characters per line. The trunk label can have up to five lines, and labels for other levels are limited to two lines.

You can split a line of text into two lines by pressing the **Return** key anywhere inside the line. Likewise, flow a line with the previous line of text by pressing the **Back space** key at the beginning of the line.

You can copy the contents of the paste buffer into a label using the PASTE command. This can be helpful when the information for your diagram is available from another source (a flat file, for example). Use the paste buffer to copy the information from that source to your Ishikawa diagram.

Some hosts designate the right mouse button for pasting, some use control keys (like ctrl-p), while others use a designated function key. For more details about using paste buffers with the SAS System, consult the SAS companion for your host.

To paste text into a label, you must first select the label. For existing arrows, select the arrow, position the cursor where you want the text to appear, and then issue the PASTE command. For new arrows pending text entry, simply issue the PASTE command. Any text in the paste buffer that causes the label to exceed its limits is truncated.

When your mouse has a paste key defined, instead of adding an arrow and pasting the text in two operations, use the *right* mouse button to add the arrow. This action adds a new arrow, automatically copies the label from the paste buffer, and terminates text entry, in a single operation.

Example

In the following diagram, the branch labeled *In-Flight Service* has been selected by clicking on the arrow tail. The arrow is highlighted with a narrow dotted line, and the text cursor is positioned over the first character in the label.

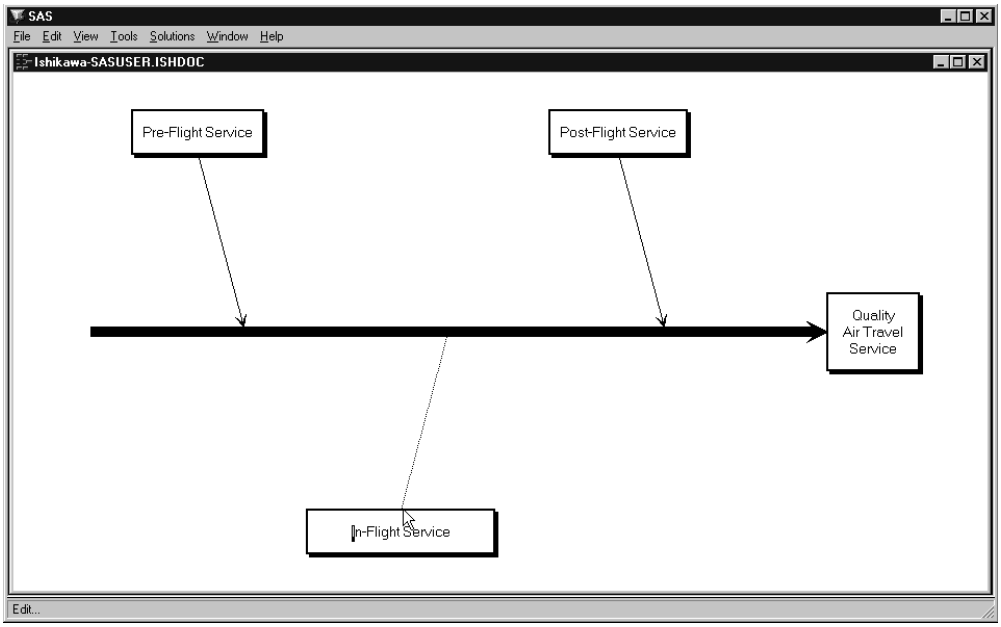


Figure 25.5. Selecting an Arrow for Editing

To change the label so that the word *Service* appears on a separate line, use the → or ← key to move the cursor to the space before the word *Service*, as shown in the following:

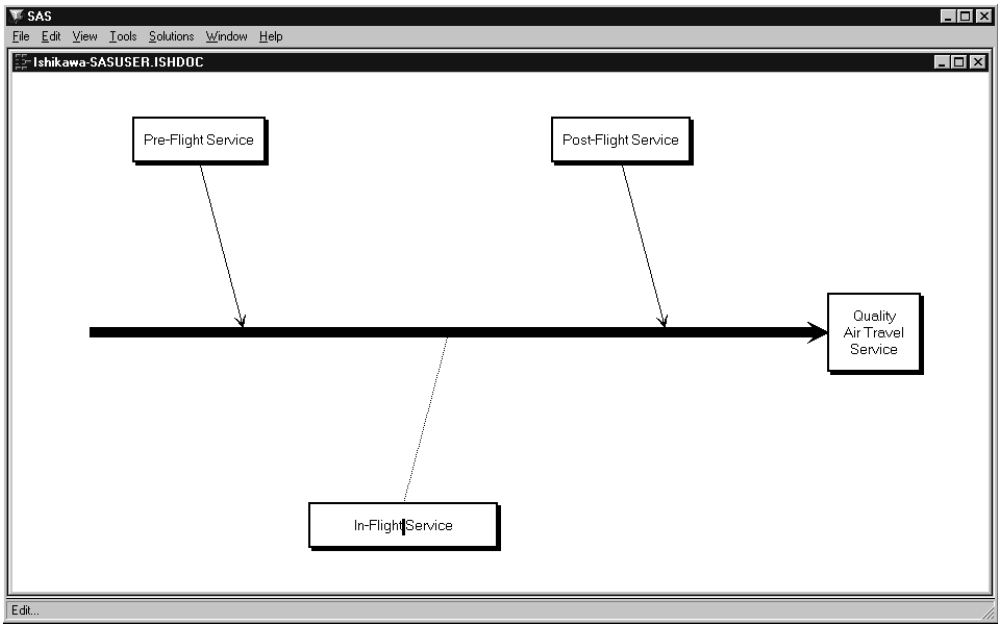


Figure 25.6. Using Cursor Keys

Now press **Return** to split the text into two lines.

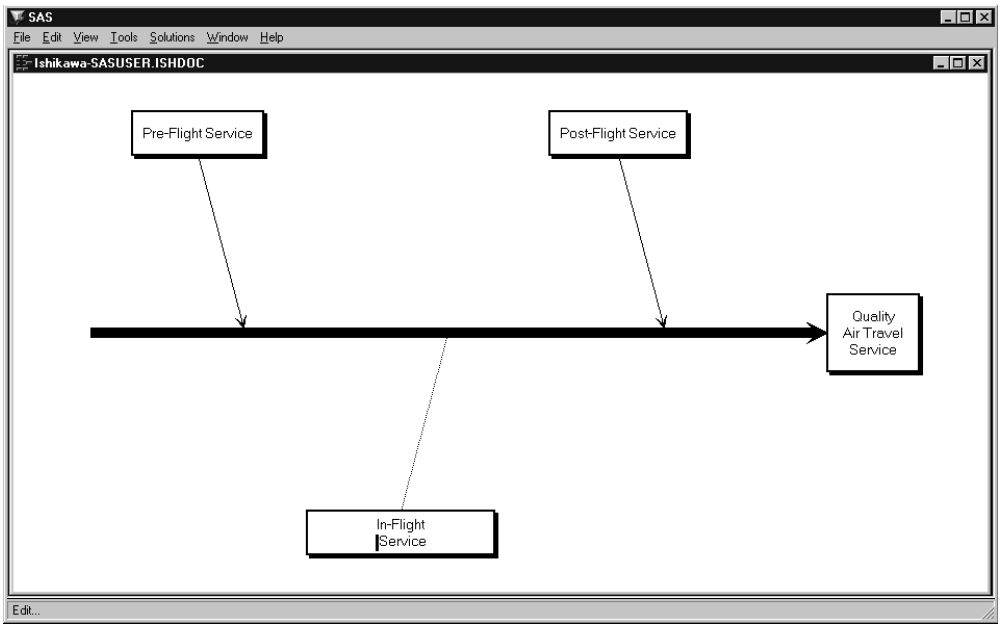


Figure 25.7. Splitting Text

Remember to delete the space preceding *Service* before pressing **Return** to terminate text entry.

Moving Arrows

You move an arrow by picking up the arrow and dropping it at a new location:

- To *pick up* an arrow, position the cursor over the arrow head and click the mouse. The arrow you selected will be highlighted with a narrow dotted line. If the arrow is not highlighted, move the cursor closer to the arrow head and repeat the click.
- To *drop* an arrow, move the cursor slightly to one side of the new attachment point and click (just as though you are adding a new arrow).

When you move an arrow, all its descendants move with it.

To cancel a move after picking up an arrow, click in the background area of the ISHIKAWA window.

Do not try to drop the arrow back into place by clicking on the arrow head a second time. A double click on (or near) the arrow head deletes the arrow. To move an arrow a short distance, move the cursor away from the arrow head before clicking to drop the arrow. On some systems the cursor will change shape when you have moved outside the context-sensitive area.

Example

As your diagrams develop, you will want to reposition arrows, either because of errors or for aesthetic reasons. The following is an example of an Ishikawa diagram that needs to be modified:

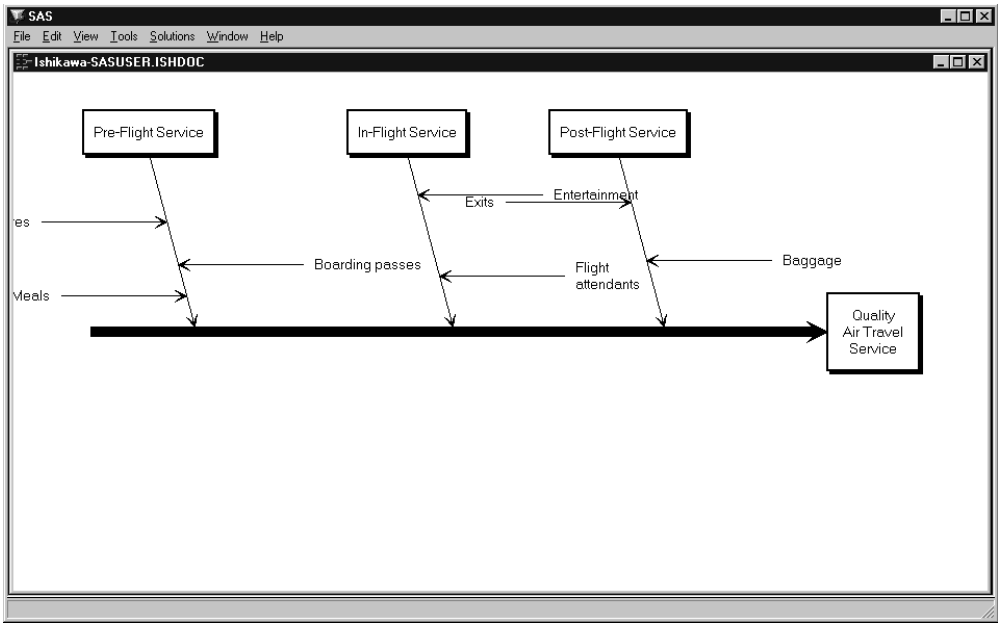


Figure 25.8. An Inelegantly Arranged Ishikawa Diagram

The diagram lacks balance, and some of the branches are too close, resulting in collisions and clipping.

One way to improve the diagram is to move the branch for *Pre-Flight Service* toward the center of the trunk. First select the arrow head for this branch.

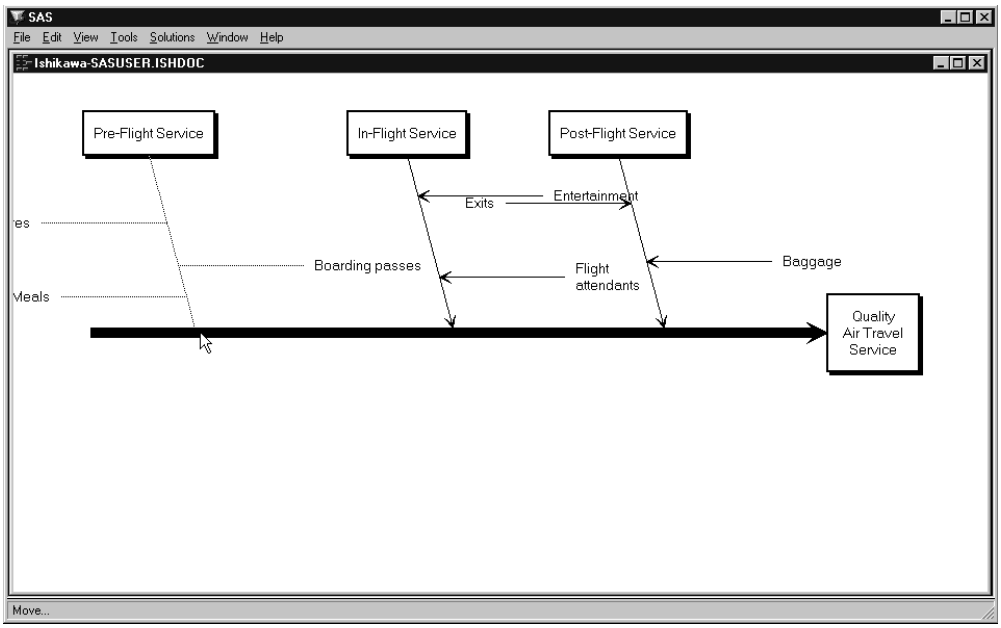


Figure 25.9. Selecting an Arrow to Move

Then move the cursor to a point just slightly above the trunk near the desired new attachment point.

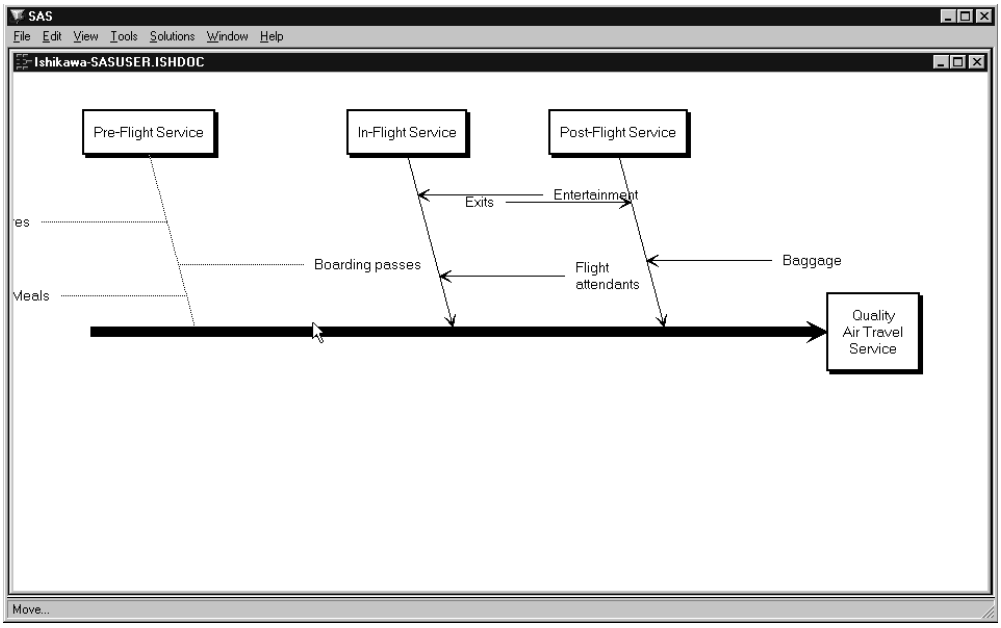


Figure 25.10. Locating the New Attachment Point

Drop the arrow in place by clicking the mouse.

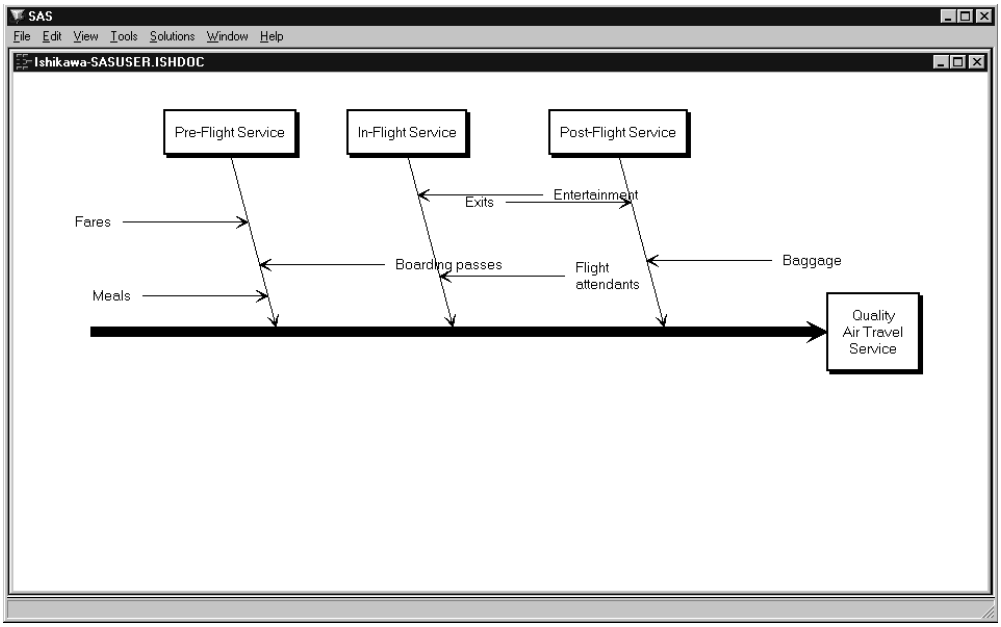


Figure 25.11. Dropping an Arrow into Position

Next, you should reflect the middle branch to the lower half of the diagram to balance the diagram and eliminate the remaining collisions. Once you have selected the branch, position the cursor slightly below the trunk near the desired new attachment point.

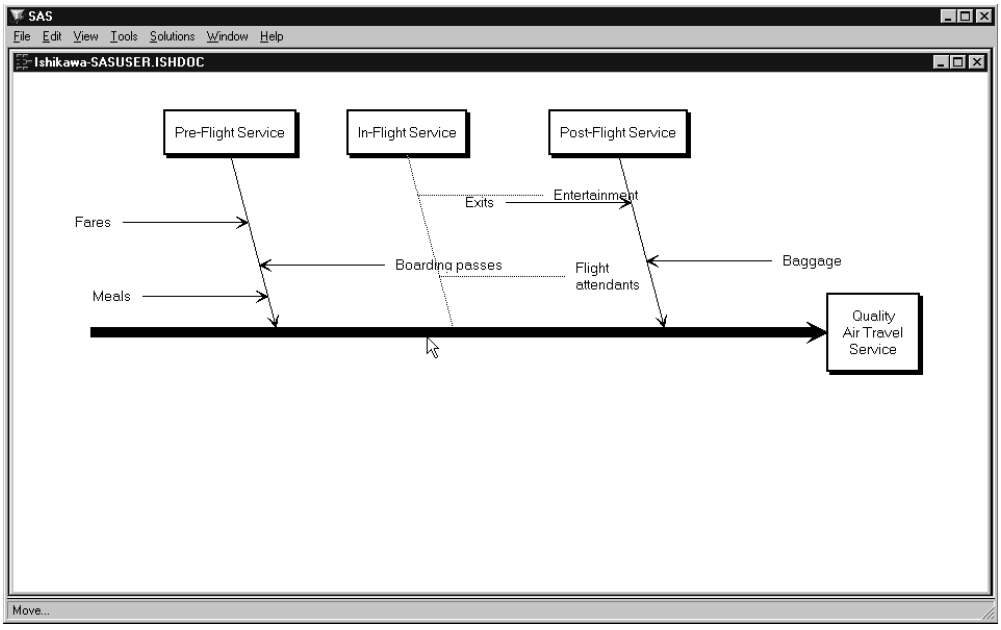


Figure 25.12. Selecting an Arrow for Reflecting

Click the mouse to complete the reflection.

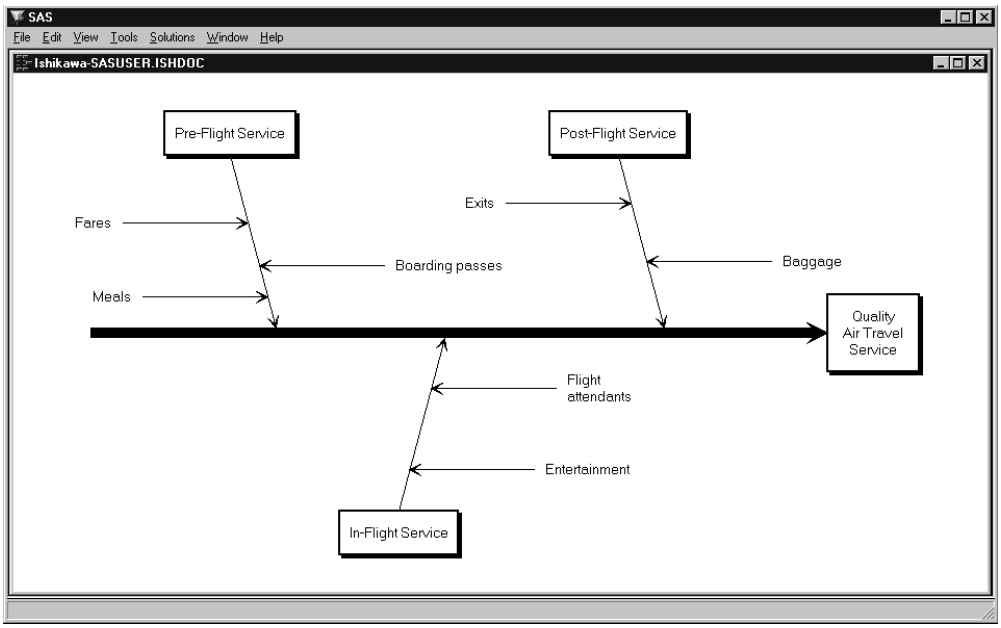


Figure 25.13. Reflecting an Arrow

Note that the stems are reflected with the branch and that their positions (relative to the trunk) are preserved.

Finally, the stem labeled *Meals* is incorrectly attached to the branch labeled *Pre-Flight Service* and should be moved to the branch labeled *In-Flight Service*. Once you have selected the stem, move the cursor slightly left of the new attachment point.

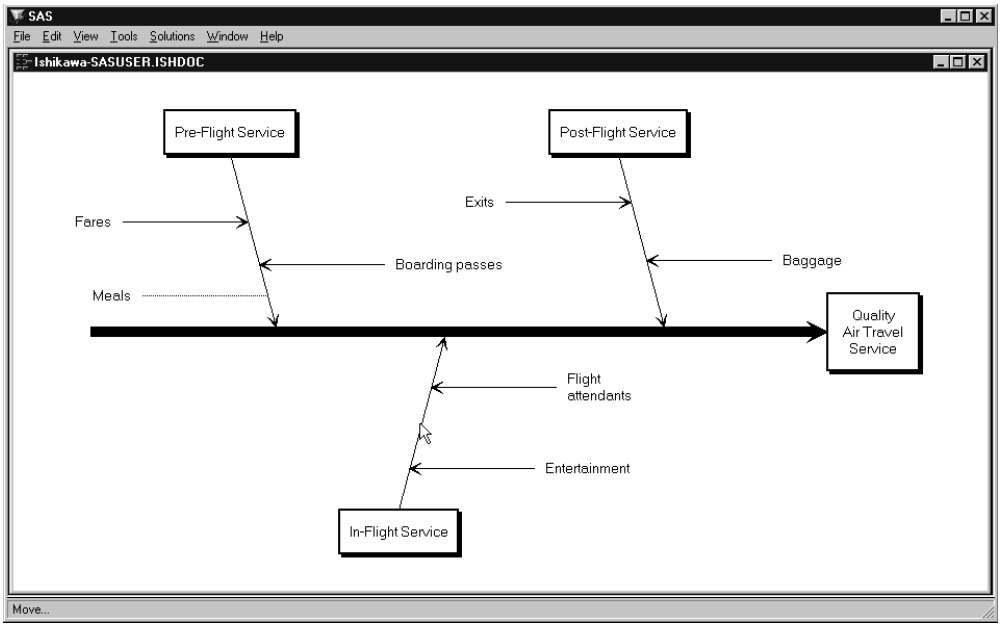


Figure 25.14. Locating the New Attachment Point

To complete the move, click the mouse.

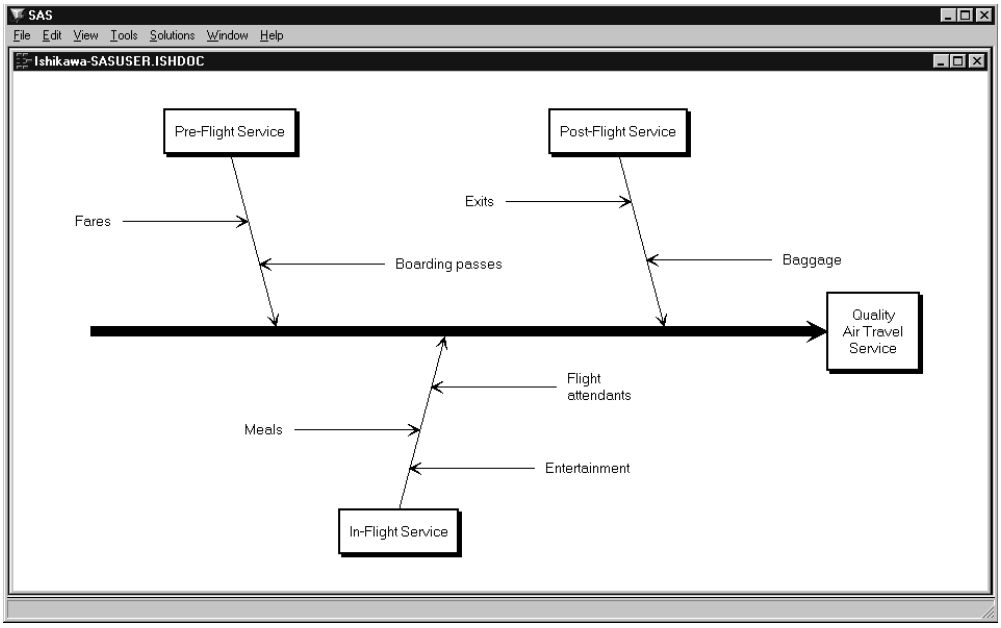


Figure 25.15. Moving a Stem

Apply the same principles when moving an arrow to a new level (for example, to elevate a stem to a branch) or a new diagram (when you have multiple ISHIKAWA windows open).

Deleting Arrows

You can delete an arrow (with all its descendants) by moving the cursor over the arrow head (attachment point) and double clicking. If you accidentally move the cursor while double clicking, it is possible that the arrow will be moved instead of being deleted. In that case, double click on the arrow head again.

You can undo a deletion by moving the cursor to a background area of the window and using the right mouse button to select **Undelete** from the background popup menu. Repeat the operation when you want to undo several deletions.

Once an arrow has been selected for deletion, you can cancel the pending operation by moving the cursor to a background area of the diagram and clicking the mouse.

The ISHIKAWA environment does not allow you to delete the trunk. To clear the window, select **Edit ▷** | **Clear...** from the command bar. Then start a new diagram by selecting **File ▷** | **New...** or **File ▷** | **Open...**.

Example

In the following diagram, the branch labeled *Post-Flight Service* has been selected for deletion (note that the branch is highlighted):

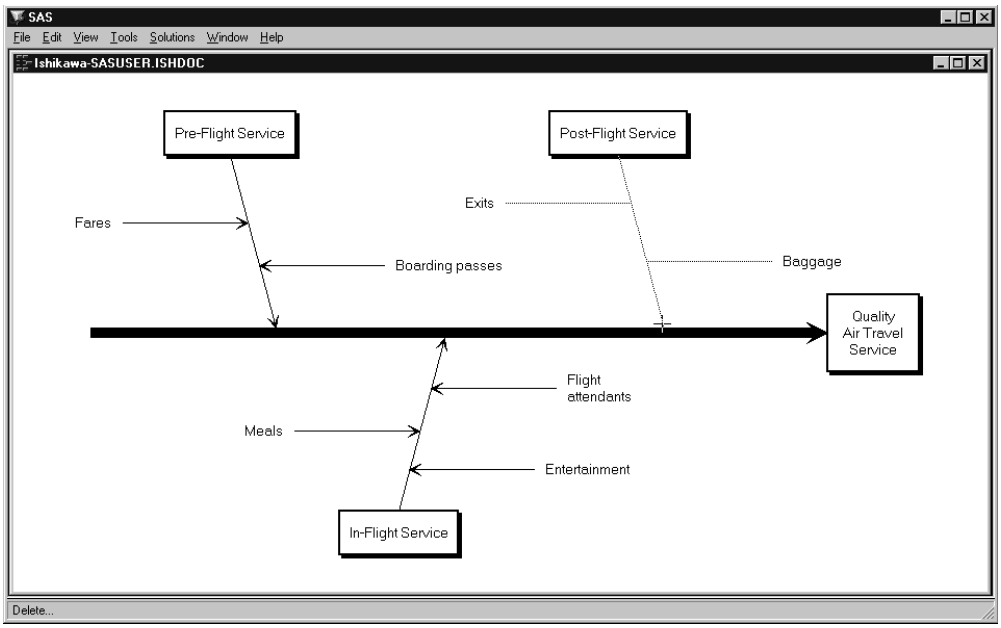


Figure 25.16. Selecting a Branch for Deletion

Without moving the cursor, click on the arrow head a second time to delete the branch.

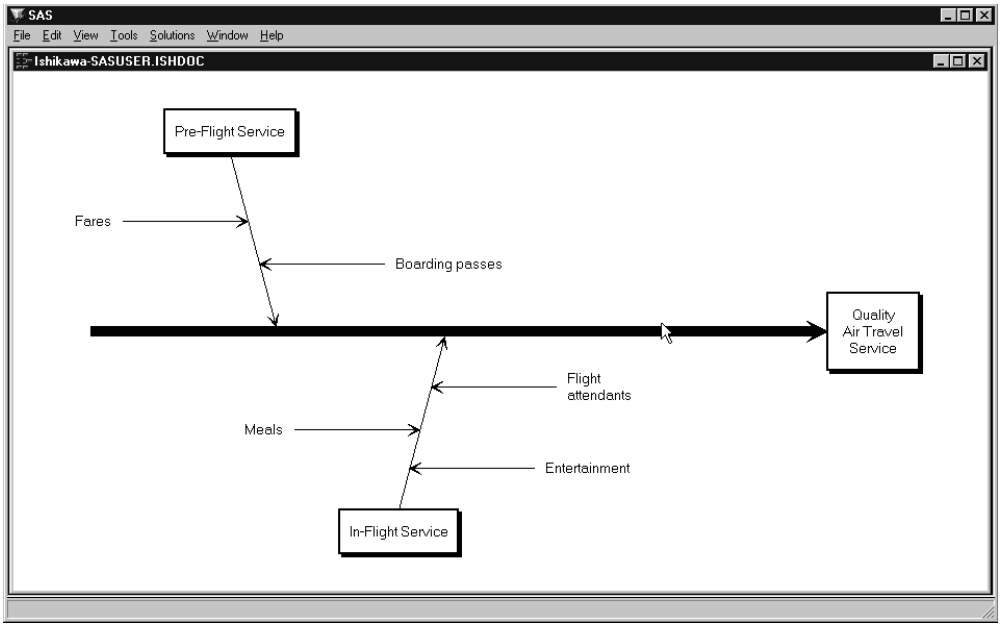


Figure 25.17. Deleting a Branch

To undelete the previous deletion, move the cursor to a background area of the window and use the right mouse button to select **Undelete** from the background popup menu.

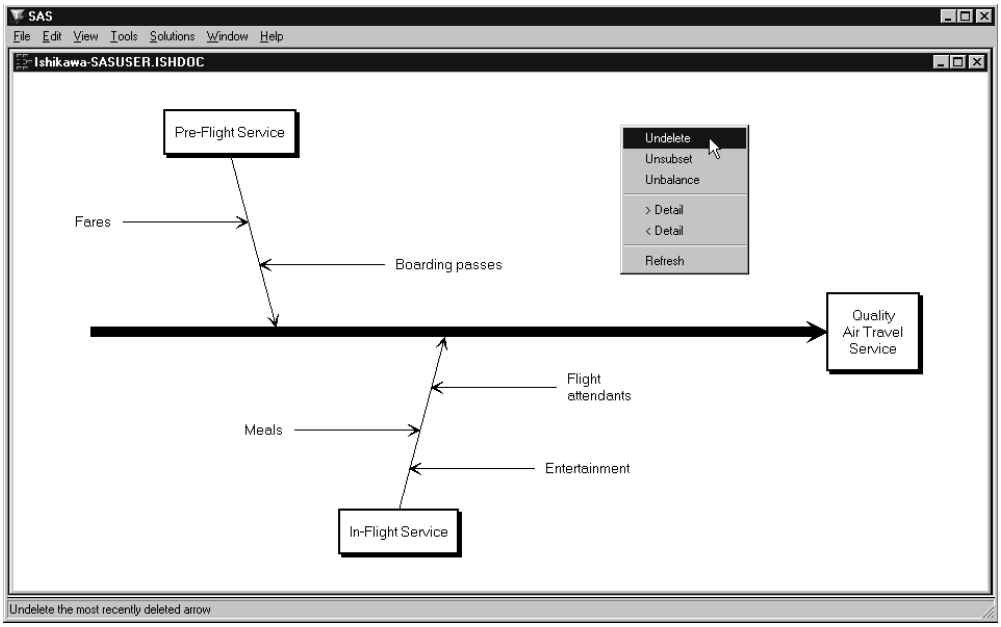


Figure 25.18. Undeleting a Branch

Resizing Arrows

You can resize an arrow by holding the mouse button down over the tail end of the arrow and dragging the mouse.* As you move the mouse, the arrow is represented by a rubberband line, and a plus sign (+) is drawn to indicate the original position of the arrow tail. The new length is determined by the position of the cursor when you release the mouse.

To cancel a resize operation once you have depressed the mouse button, release the button outside the ISHIKAWA window.

All non-horizontal arrows are constrained to have the same angle. You control the angle by resizing a branch. That is to say, when you resize a leaf, its angle does not change.

Use **View ▷** | **Ishikawa Settings ▷** | **Resize Method ▷** to control the scope of the resizing operation.

- **Local** resizes only the arrow being dragged.
- **Global** resizes all the arrows at that level to lengths that are proportional to the arrow being dragged. This is the default.
- **Uniform** resizes all arrows at that level to the length of the arrow being dragged.

*Some devices (such as the IBM3179) require you to define a drag key. For more details about dragging on your system, consult the SAS companion for your host.

When you resize an arrow, you also update the default size for all new arrows at that level.

By default, global and uniform resizing applies to all the arrows at the level of the arrow being resized. To restrict resizing to a specific subset of arrows, you can subset them as follows:

- Move the cursor over the arrow head of an arrow to subset that arrow and all its descendants.
- Move the cursor over the arrow tail of an arrow to subset only that arrow (and not its descendants).
- Use the right mouse button to activate the popup menu.
- Select **Subset**.

On some hosts, shift-clicking on the arrow head or tail also subsets an arrow.

Subsetted arrows are indicated by underlined labels. Subsetting is a toggle operation, so to *unsubset* an arrow, repeat the preceding steps.

To unsubset all the arrows in the diagram, do the following:

- Move the cursor to a background area of the window.
- Use the right mouse button to activate the background popup menu.
- Select **Unsubset**.

Be sure to remove all subsets after you have finished modifying the diagram, since remaining subsets can alter the focus of other operations.

See "Modifying Arrow Colors and Line Styles" on page 729, for more examples of how subsets are used.

Example

Arrows that are too long can cause clipping and collisions, as illustrated in the following diagram:

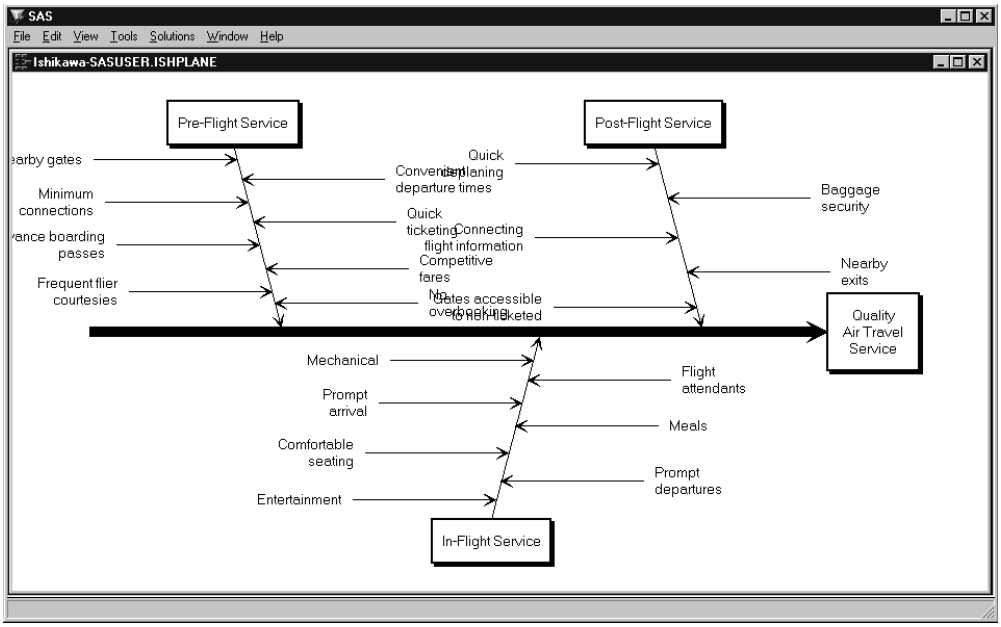


Figure 25.19. Before Resizing the Diagram

To resize the stems in the upper half of the diagram, proceed as follows:

- Subset the branch for *Pre-Flight Service* by moving the cursor over its arrow head and selecting **Subset**.
- Do the same to *Post-Flight Service*.
- Shorten one of the subsetted stems by dragging its tail to the desired length.
- Remove all subsets by selecting **Unsubset**.

The results are as follows:

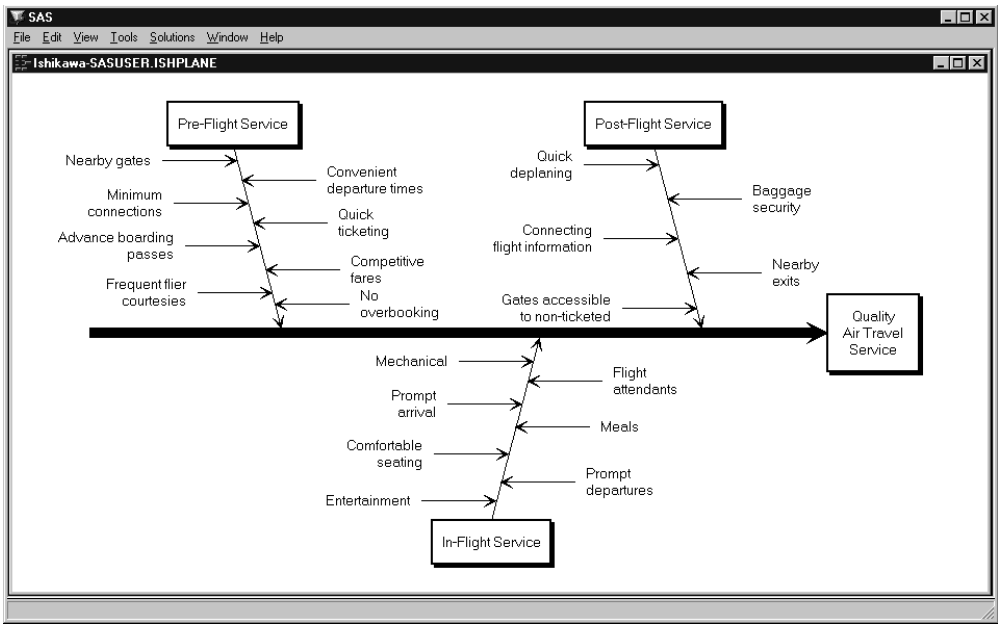


Figure 25.20. After Resizing the Diagram

Swapping Arrows

Use the swap operation to interchange two arrows in a single operation instead of using two move operations. Swapping has all the flexibility of the move operation; you can swap arrows that have different parents, different levels, or arrows from different diagrams.

Like moving, the results depend upon whether you select the arrow from the arrow head or the arrow tail. When you select the arrow head, the arrow and all its descendants are moved. When you select the arrow tail, only the labels of the selected arrows are interchanged.

Swapping is a two step operation.

- Move the cursor over the arrow head (tail) of one of the arrows to be swapped and select **Swap** from the context-sensitive popup menu.
- Complete the swap by using the mouse to select the comparable end (head or tail) of the second arrow.

To cancel a swap after you have selected the first arrow, click in a background area of the diagram.

Example

To swap the branch labeled *Pre-Flight Service* (and all its descendants) with the branch labeled *Post-Flight Service* in the following diagram, move your cursor over

the arrow head of the *Pre-Flight Service* branch and activate the popup menu using the right mouse button. Select **Swap** to begin the operation.

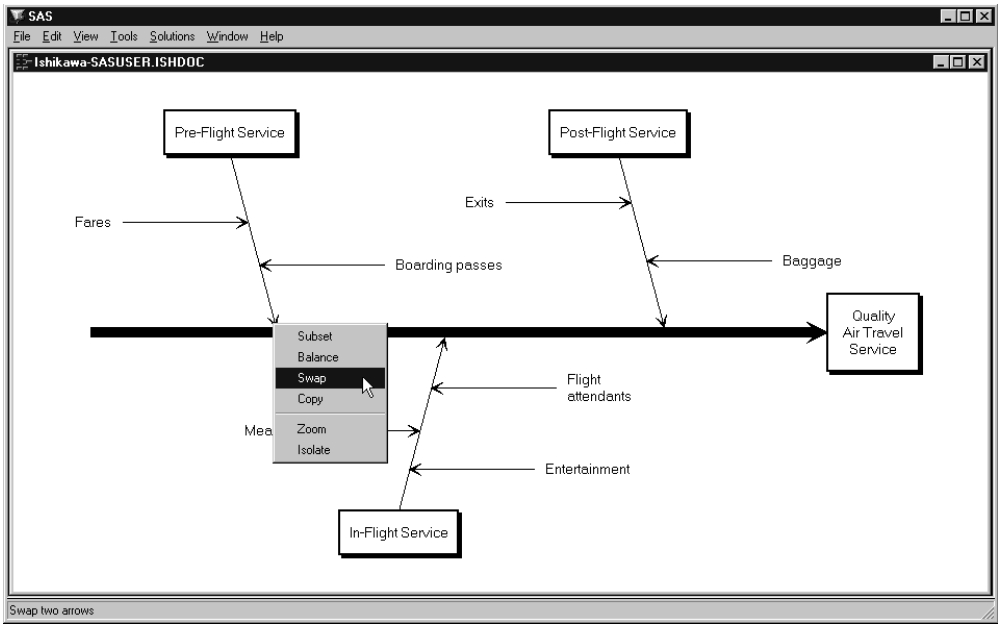


Figure 25.21. Swapping Two Arrows

To complete the swap, select the arrow head of the *Post-Flight Service* branch.

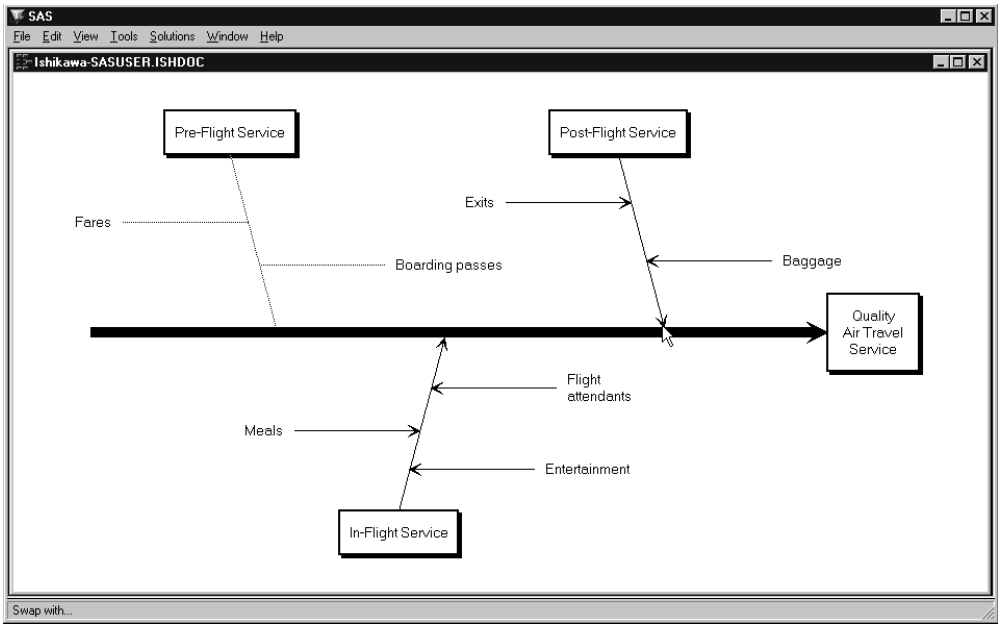


Figure 25.22. Swapping Two Arrows *(continued)*

The completed diagram illustrates how the swap operation simplifies interchanging

two arrows.

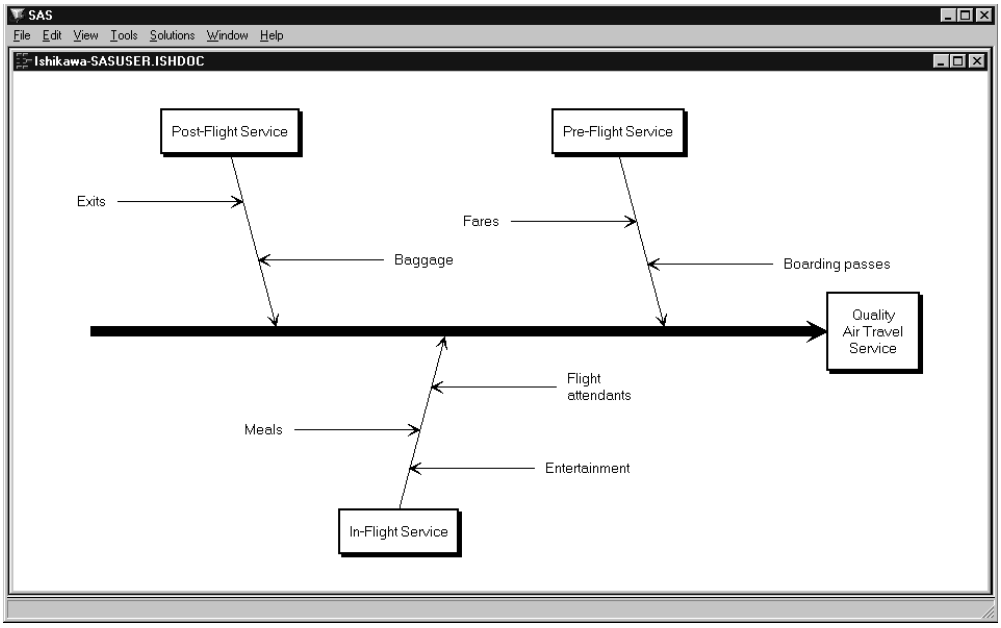


Figure 25.23. Completing a Swap

An alternative to swapping the arrows is to move them. However, moving arrows in this situation requires more steps and tends to be more cumbersome than swapping.

Balancing Arrows

An Ishikawa diagram is said to be *balanced* if the sub-arrows attached to each arrow are equally spaced.

To balance the immediate descendants of an arrow *and all its descendants*, proceed as follows:

- Move the cursor over the arrow head.
- Activate the popup menu using the right mouse button.
- Select **Balance**.

To balance only the immediate descendants of an arrow, select **Balance** from the popup menu for the arrow tail.

You can restore the arrows to their original positions by doing the following:

- Activate the background popup menu using the right mouse button.
- Select **Unbalance**.

The ISHIKAWA environment provides three alternative methods for balancing arrows. Select one of the following choices from the **View** ▷ | **Ishikawa Setting** ▷ | **Balance Method** ▷ menu:

- **Preserve order/sides** maintains the order and directions of the sub-arrows but repositions them so they are evenly spaced.
- **Preserve order/alternate sides** maintains the ordering of the arrows but repositions adjacent arrows so that they appear on opposite sides. This is the default.
- **Preserve sides** maintains the side on which the sub-arrows are attached then spaces each side of the arrow independently.

Example

Consider the following unbalanced diagram:

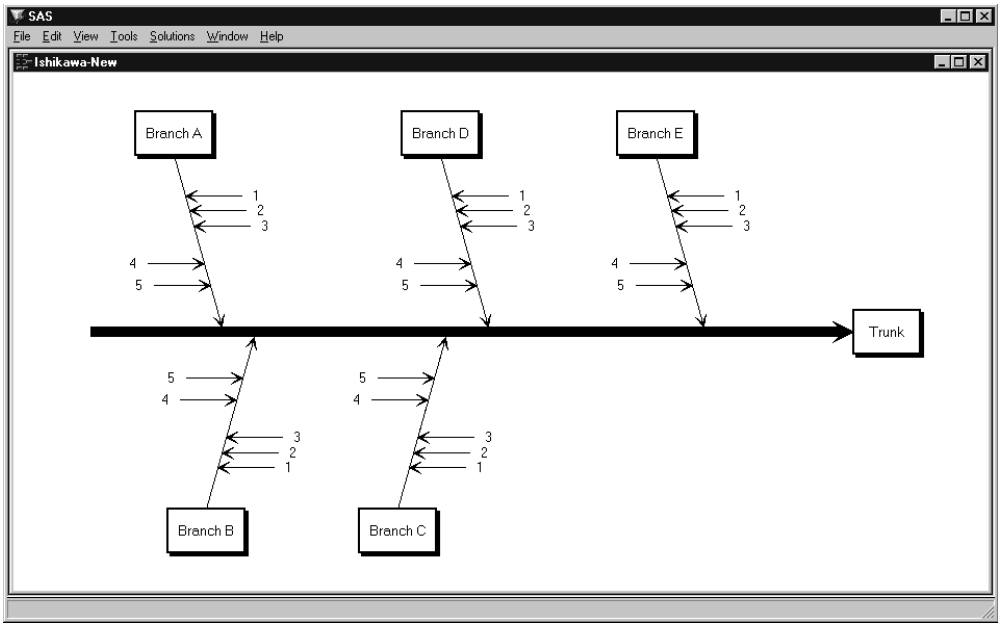


Figure 25.24. An Unbalanced Ishikawa Diagram

To balance only the stems of the branch labeled *Branch A*, move the cursor over the arrow head and press the right mouse button.

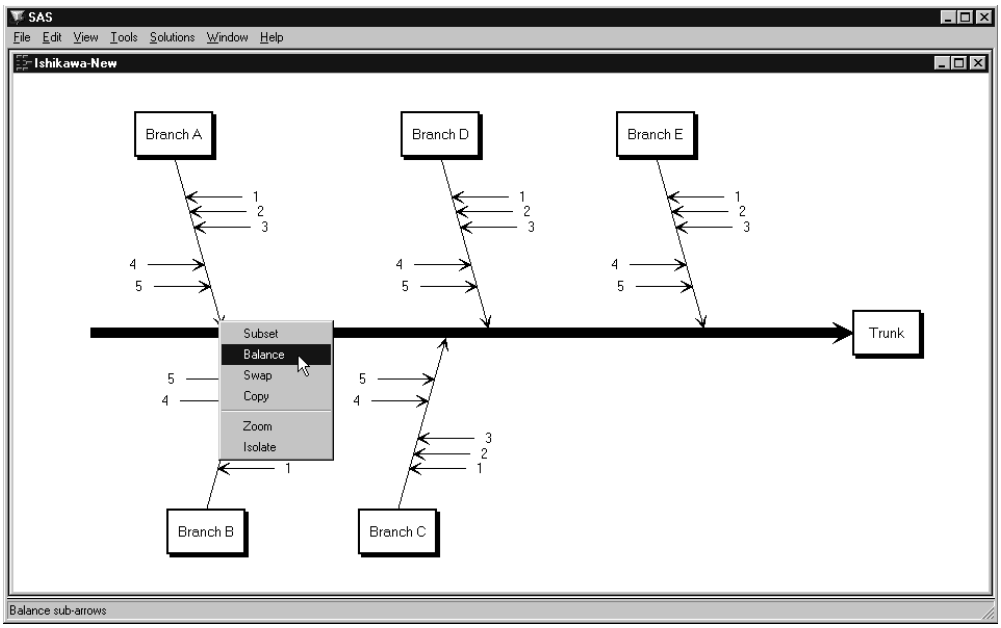


Figure 25.25. Balancing a Branch

Select **Balance** from the arrow head popup menu.

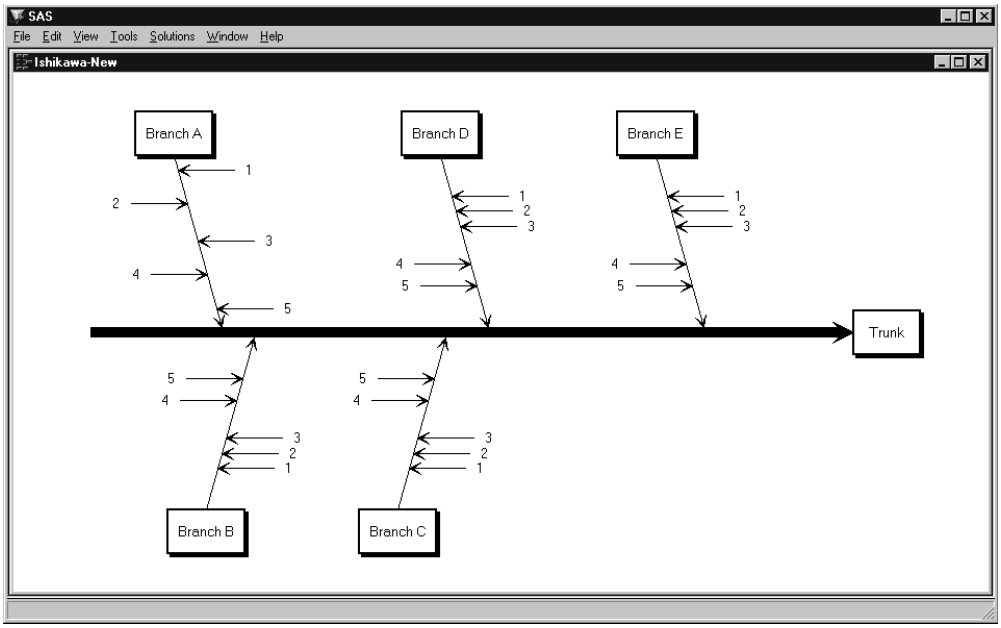


Figure 25.26. A Balanced Branch

Note that since the stems are without leaves, selecting either the head or the tail has the same result.

To balance only the five major branches in the preceding diagram without affecting

their stems, move the cursor to the tail end of the trunk and select **Balance** from the popup menu.

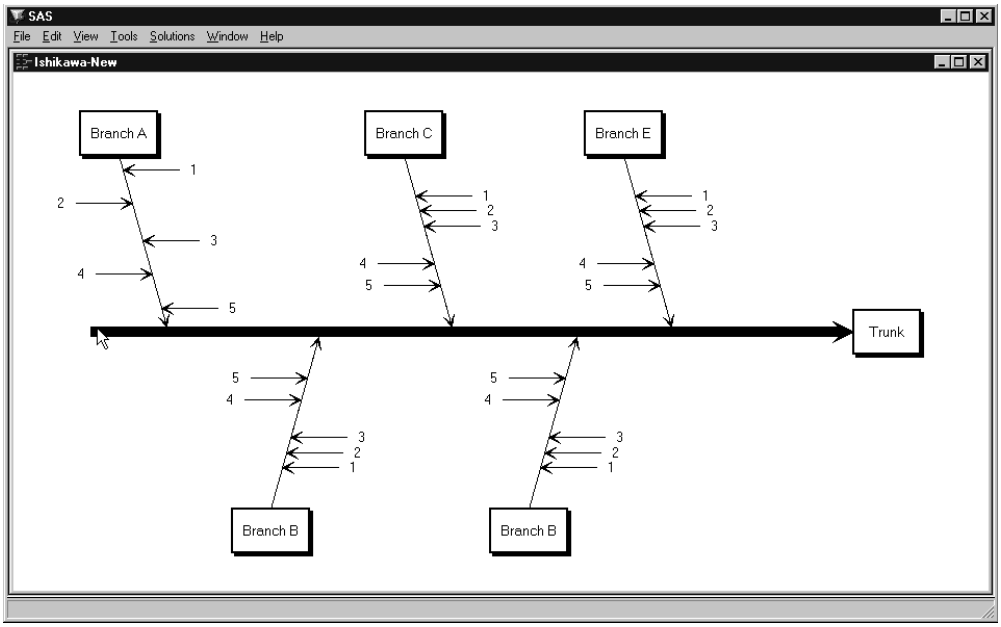


Figure 25.27. Balancing Only the Branches

To balance the entire diagram (from head to tail, so to speak), move the cursor to the head of the trunk and select **Balance** from the popup menu.

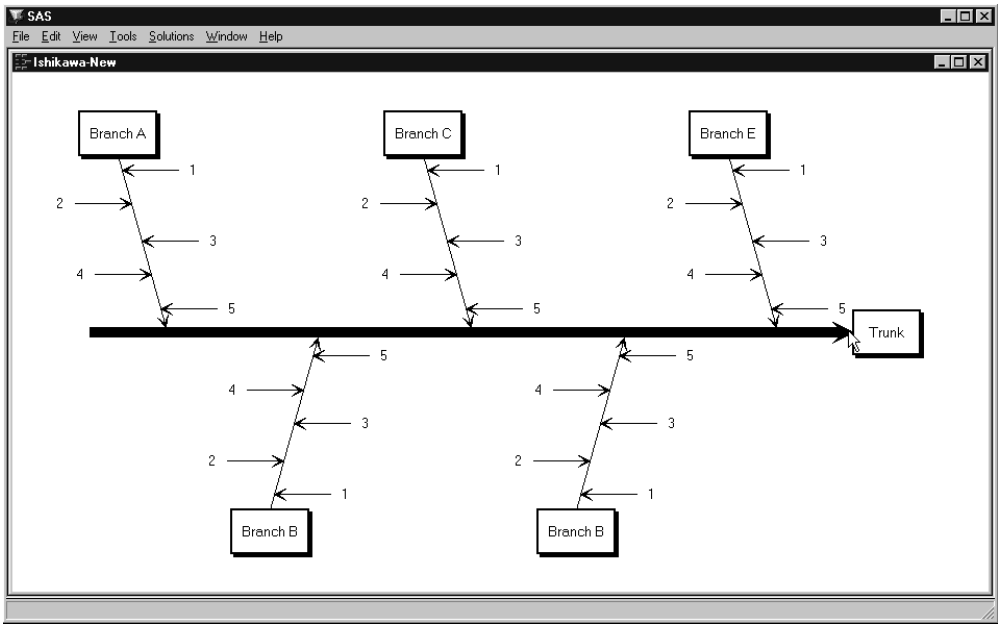


Figure 25.28. Balancing the Entire Diagram

Note that the balancing method used here not only changes the spacing of the stems but reflects them as needed to achieve a balanced appearance. You can control this by specifying a balancing method, as illustrated by the next example.

Example

The following diagram displays an unbalanced branch and a copy of that branch after it was balanced using the **Preserve order/alternate sides** balancing method:

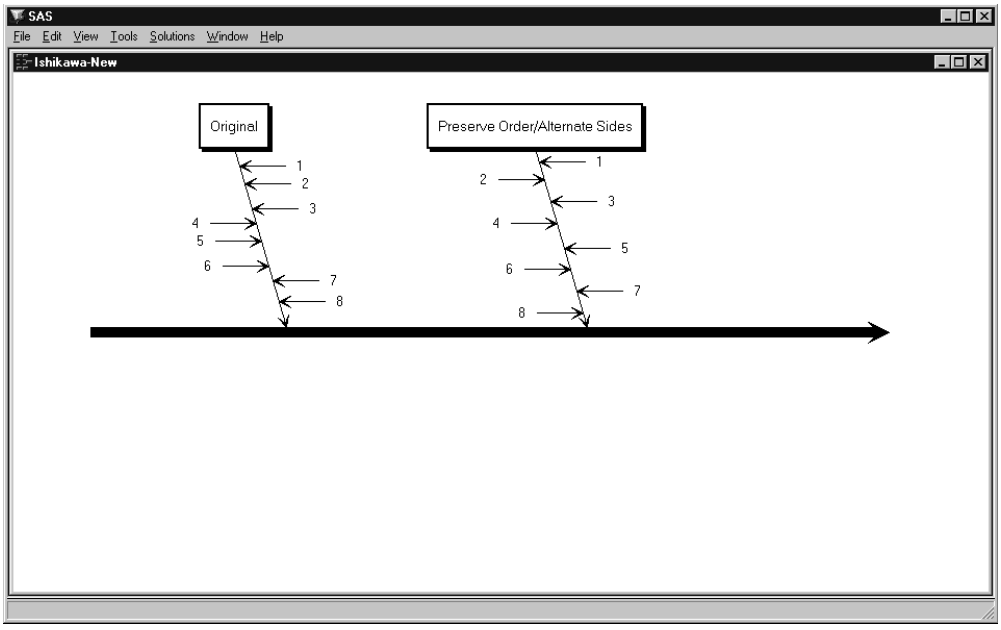


Figure 25.29. Preserving Order But Alternating Sides

Note that the stems remain in order (*1*-*8*) from tail to head, but they now alternate evenly across both sides of the branch. This is the default method used for balancing arrows.

Example

The following diagram displays an unbalanced branch and a copy of that branch after it was balanced using the **Preserve order/sides** method:

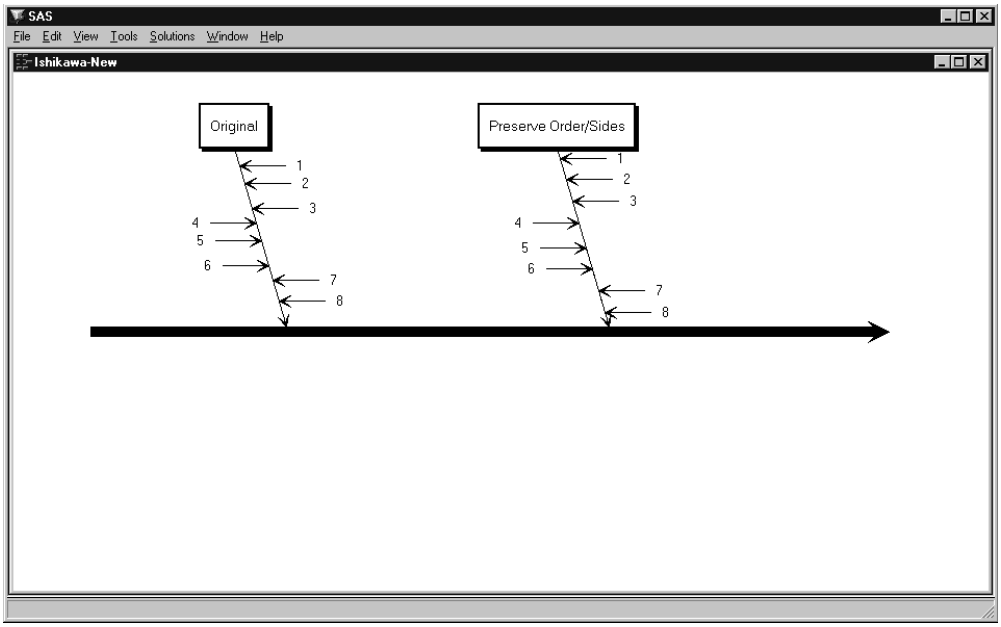


Figure 25.30. Preserving Order and Sides

Note that stems *4-6* remain on the left, stems *1-3* and *7-8* remain on the right, and the order from tail to head is still *1-8*. However, the stems are now spaced uniformly.

Example

The following diagram displays an unbalanced branch and a copy of that branch after it was balanced using the **Preserve sides** balancing method:

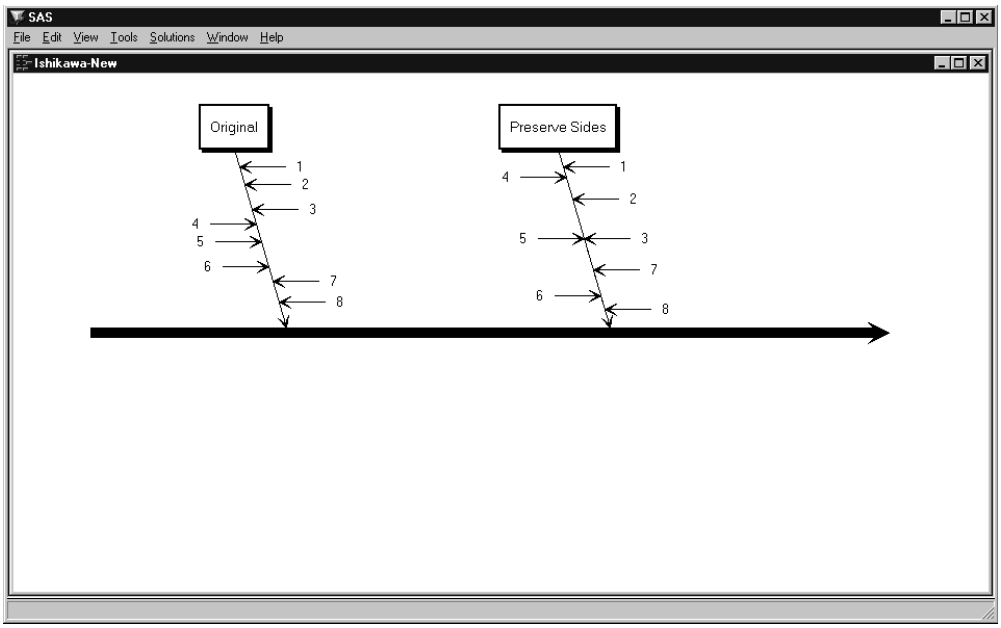


Figure 25.31. Preserving Sides

Note that the stems on the left (*4-6*) are spaced uniformly, and the stems on the right (*1-3* and *7-8*) are spaced uniformly. The two sides are spaced independently of each other.

Notepads

Ishikawa (1982) and Kume (1985) advocate the display of quantitative information with the arrows in an Ishikawa diagram.

In the ISHIKAWA environment, you can use *Notepad* windows to record or display information associated with each arrow. To open the Notepad window, move the cursor over the arrow tail and double click.

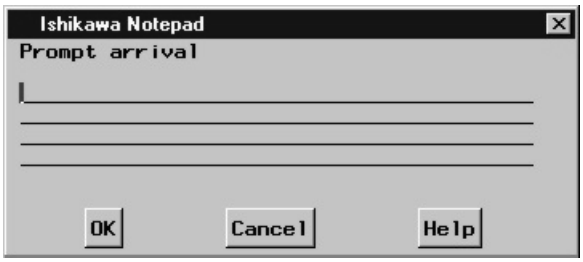


Figure 25.32. Ishikawa Notepad

Notes are limited to four lines of text with no more than 40 characters per line.

When you save your Ishikawa diagram, your notes are saved with the SAS data set.

Later, when you retrieve your diagram, all the notes are restored.

You must close the *Notepad* window before you continue working in the ISHIKAWA environment.

Example

In the following figure, double clicking on the *Prompt arrival* stem reveals details about prompt arrival times:

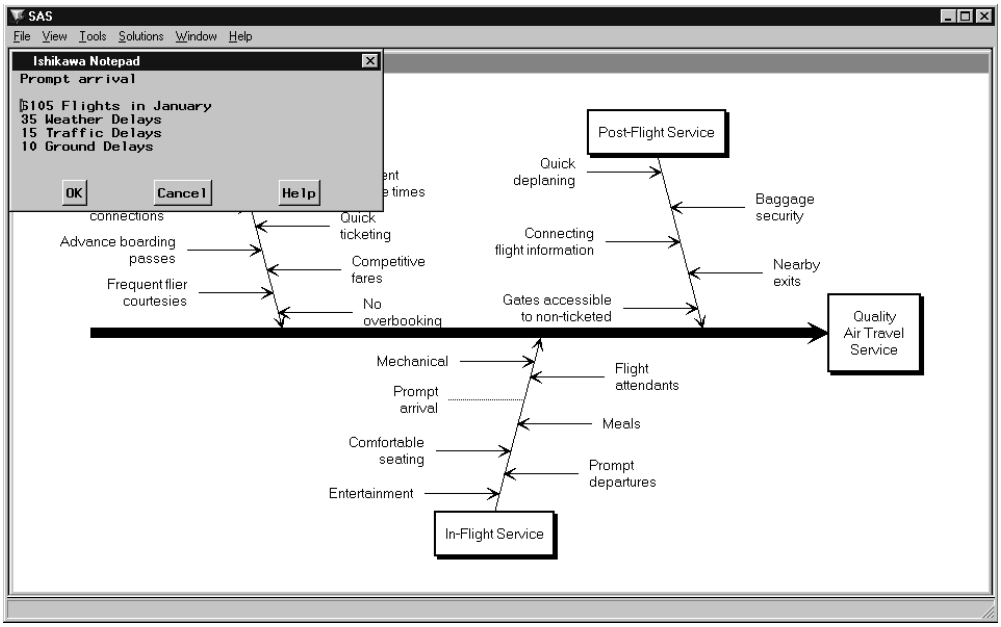


Figure 25.33. Using Notepads to Organize Details

Managing Complexity

A major advantage of the ISHIKAWA environment is that you can quickly organize a highly complex diagram. However, not everyone may be interested in seeing all the details—at least initially.

To increase the level of detail by one level, do the following:

- Move the cursor to a background area of the window, and use the right mouse button to activate the background popup menu.
- Select **> Detail**. On some hosts, you can press the **>** key instead of using the popup menu (as long as you are not editing text).

Each time you select **> Detail** from the background popup menu, the detail increases by one level.

To reverse the process and decrease the level of detail, select **< Detail** from the popup menu, or press the **<** key.

Example

You are making an online presentation about factors that influence the quality of air travel service. The following diagram presents too many details to be a good starting point for your audience:

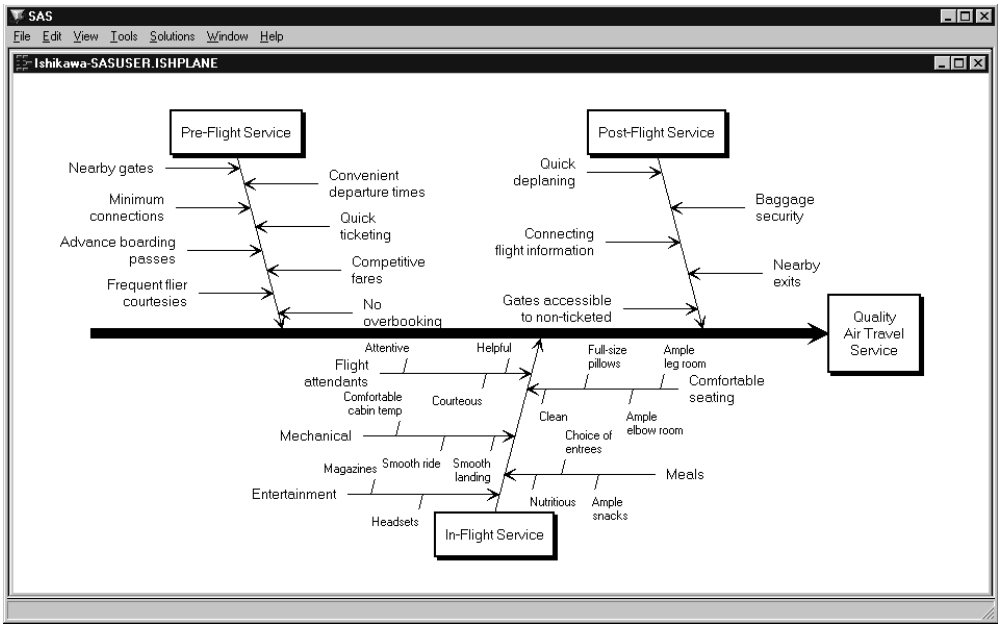


Figure 25.34. Highly Detailed Ishikawa Diagram

A better way to begin is by displaying only the trunk and branches.

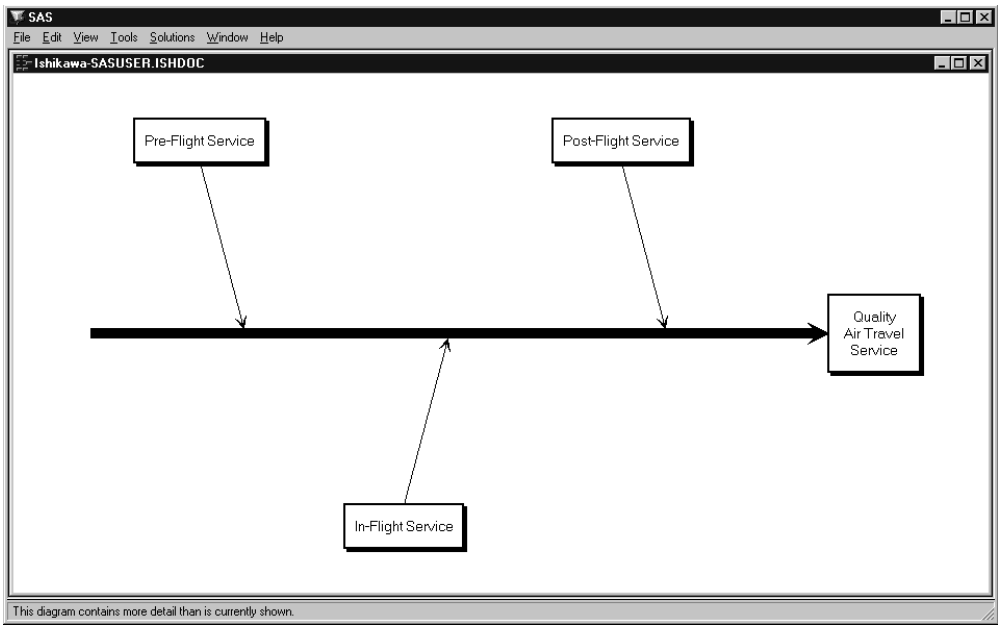


Figure 25.35. Branch-Level Diagram

Then, at the next stage of your presentation, dynamically foliate the branches with stems, as follows:

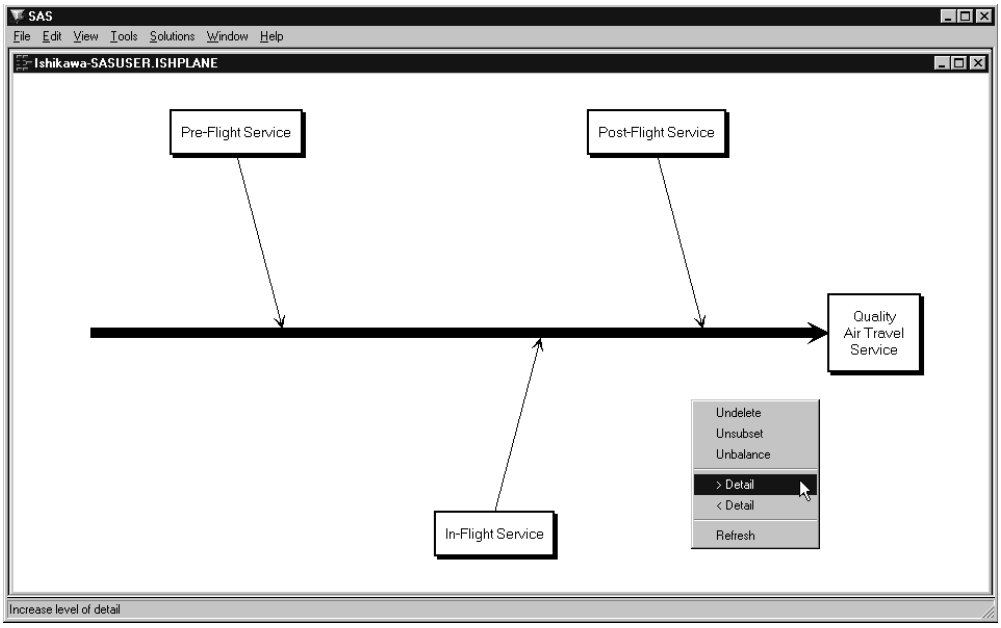


Figure 25.36. Increasing the Level of Detail

The amount of detail is increased by one level.

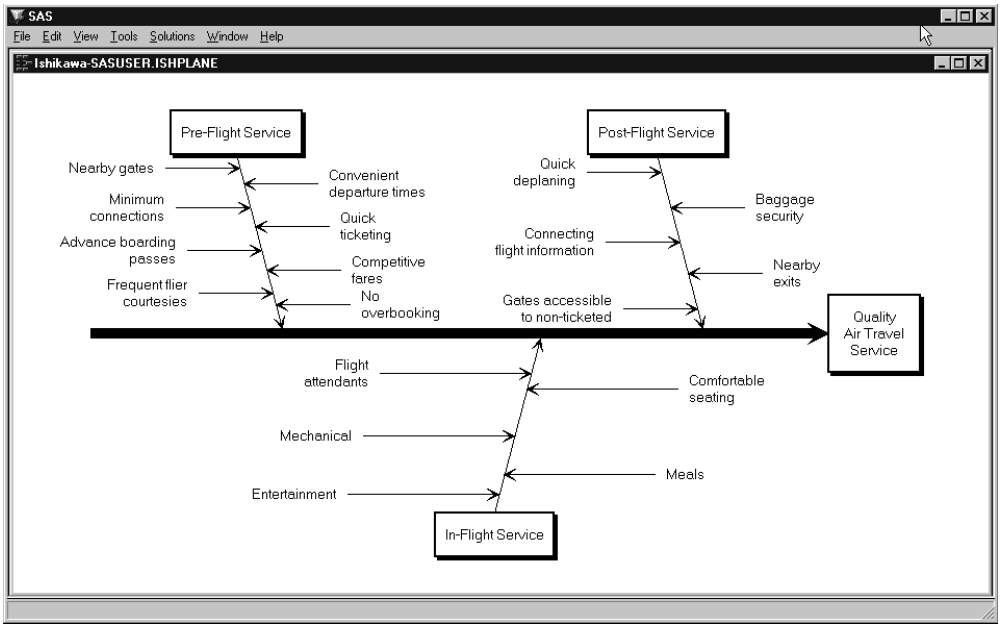


Figure 25.37. Increasing the Level of Detail

Zooming Arrows

A second method for managing a highly detailed Ishikawa diagram is to work with a subsection of the diagram in a separate window. The window and the sub-arrows inside it can be resized independently of the parent window. In all other respects, the information in the two diagrams is linked dynamically. Changes in one window (for instance, moving, adding, and editing arrows) are reflected in the other window.

To zoom an arrow, proceed as follows:

- Move the cursor over the arrow head.
- Activate the popup menu using the right mouse button.
- Select **Zoom**.

To return or *unzoom*, select **File** ▷ **Close**.

You can have up to four windows open at one time.

To reduce the amount of window management, you can specify that zoomed diagrams are to be displayed in the current window rather than in new windows by setting **Zoom Window** to *Current* in the **View** ▷ **Ishikawa Settings** ▷ **Other...** dialog.

Example

The following figure shows a branch labeled *In-Flight Service* after it has been zoomed into a new window:

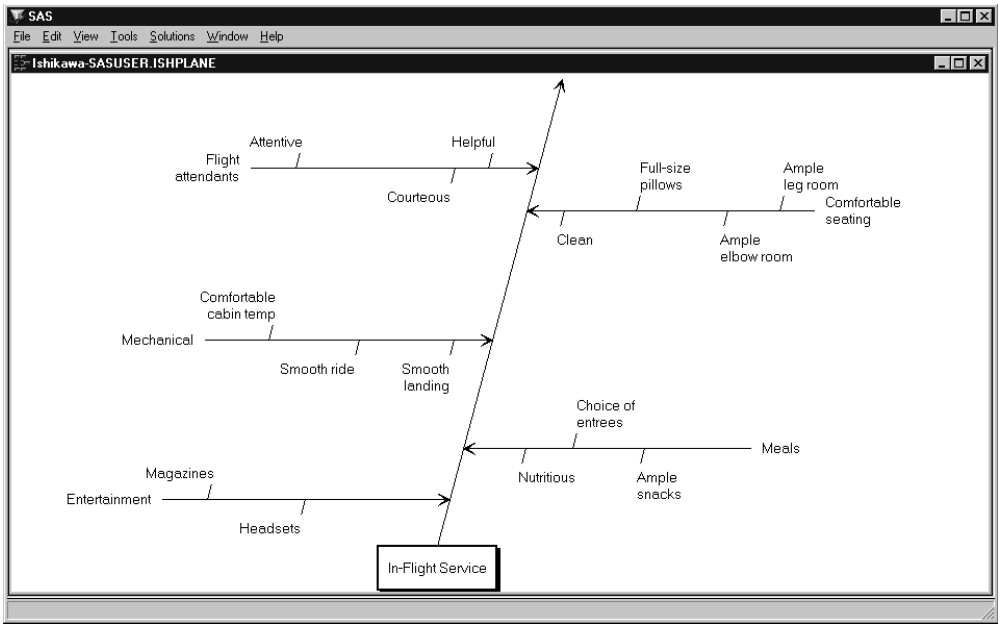


Figure 25.38. Zooming a Branch

Isolating Arrows

A third method for managing a highly complex Ishikawa diagram is to view the entire diagram as a collection of smaller diagrams. Any arrow (along with its sub-arrows) can be isolated into a separate diagram in a new window. This diagram can then be easily saved in a separate file.

To isolate a branch as a separate diagram, do the following:

- Move the cursor over the head of the arrow.
- Activate the popup menu using the right mouse button.
- Select **Isolate**.

You can have up to four ISHIKAWA windows open at one time.

Example

Consider the following diagram:

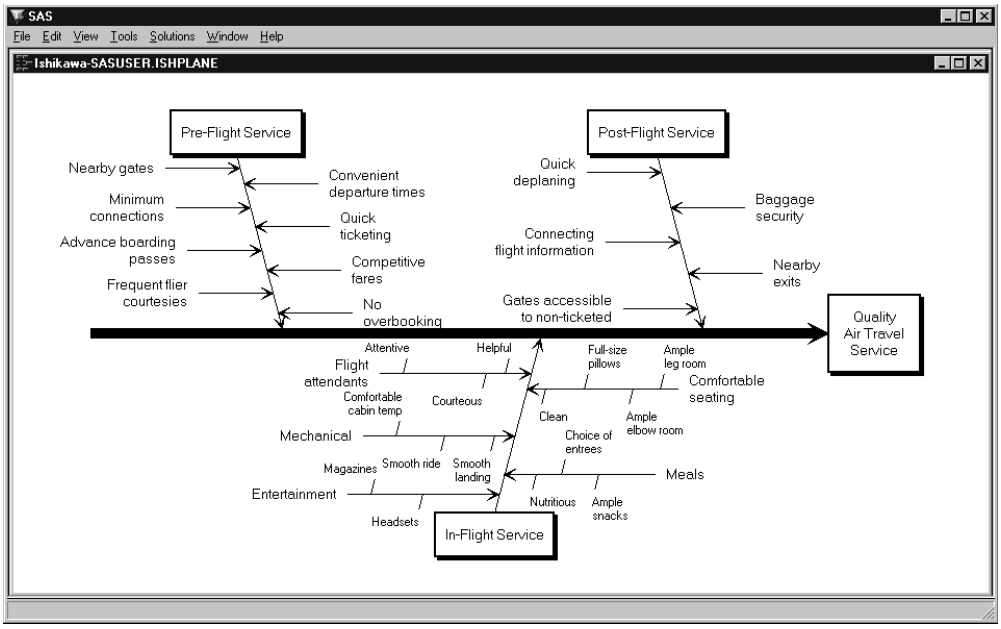


Figure 25.39. A Highly Detailed Diagram

To isolate the branch labeled *In-Flight Service* as a separate Ishikawa diagram, move your cursor over the head of the arrow. Use the right mouse button to activate the popup menu and select **Isolate**.

The following figure shows the main diagram in one window and the branch labeled *In-Flight Service* after it has been isolated to another window:

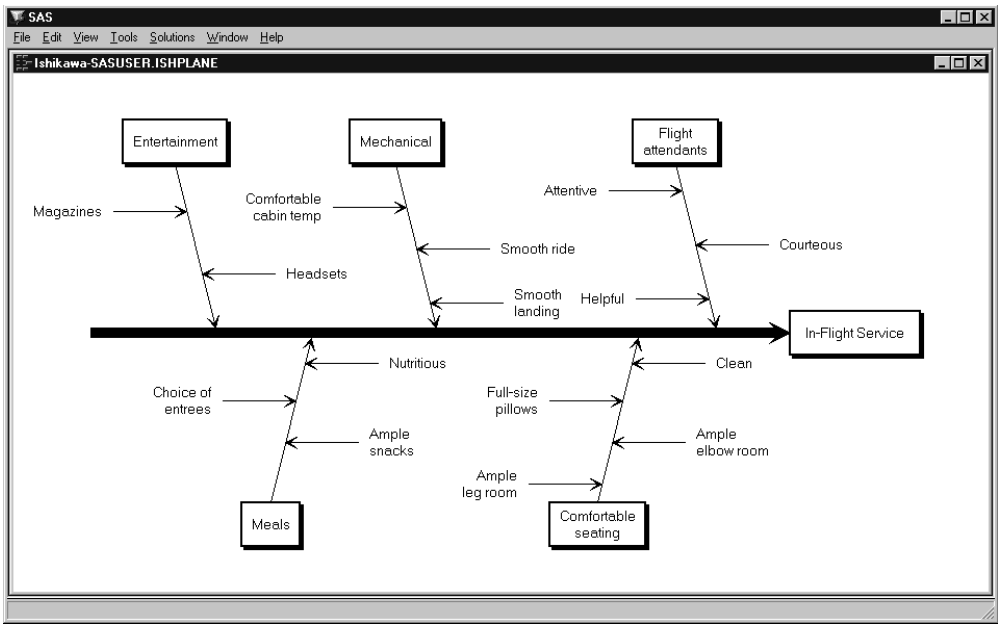


Figure 25.40. Promoting a Branch into a New Diagram

To return to the original diagram, select **File** ▷ | **Close** |.

Merging Diagrams

You can combine multiple Ishikawa diagrams into a *master* diagram by using the merge operation. To merge a stored diagram into the current diagram, proceed as follows:

- Select **File** ▷ | **Merge** |.
- Specify the name of a SAS data set that contains a saved Ishikawa diagram.

Another way to combine diagrams is to open separate ISHIKAWA windows for each sub-diagram then copy them into the master diagram. To copy all or part of an Ishikawa diagram from one window to another, do the following:

- Move the cursor over the head of the arrow.
- Activate the popup menu using the right mouse button.
- Select **Copy** |.
- Position the cursor slightly to one side of the new attachment point and click (just as though you are adding a new arrow).

Example

Suppose you want to create the following diagram by combining information from diagrams already created by each of the major service areas (Pre-Flight, In-Flight, and Post-Flight) and stored in different SAS data sets:

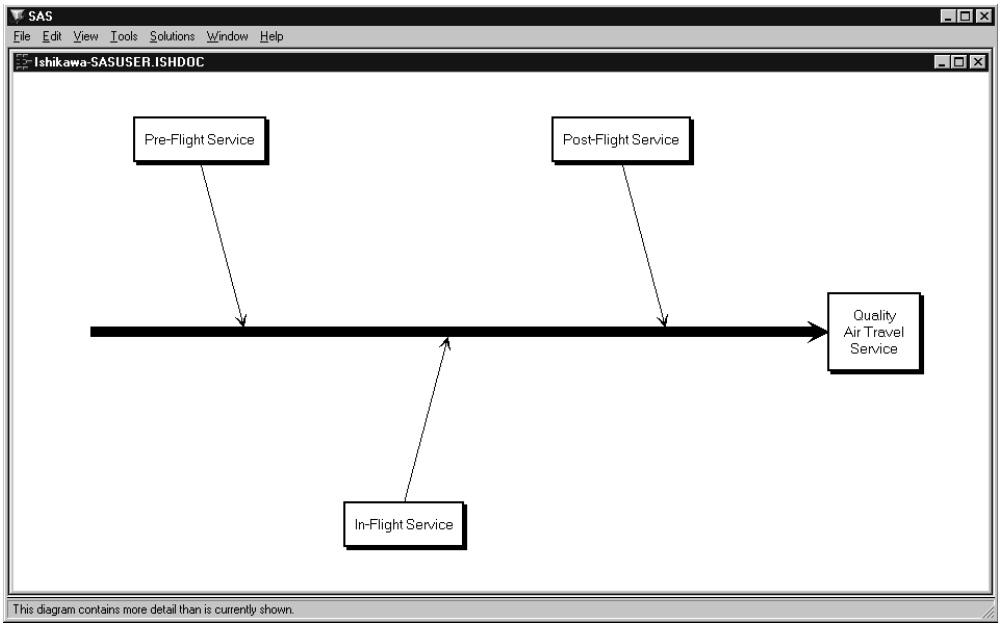


Figure 25.41. A Completed Master Diagram

First, use the ISHIKAWA environment to create the trunk for the new master diagram.

Figure 25.42. Starting a Master Diagram

Select **File** ▷ **Merge** from the command bar to open the File Requestor dialog.

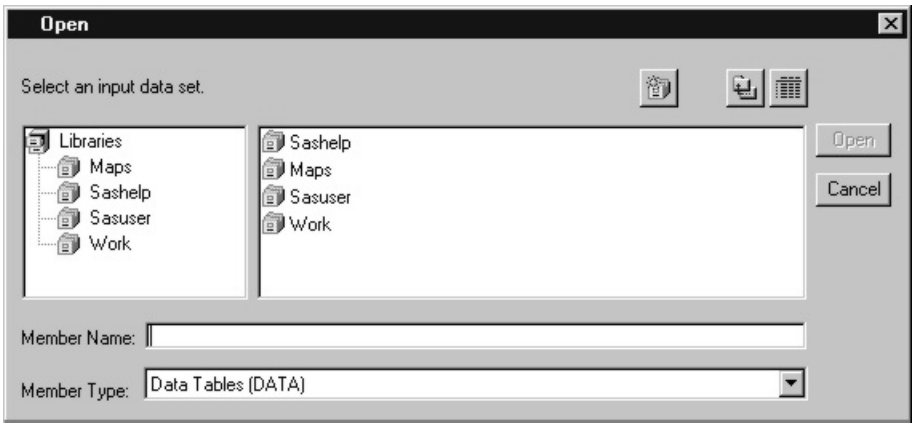


Figure 25.43. Member Selector

Specify the name of the data set for *Pre-flight services* and press **Open**.

Now click on a point along the trunk where this sub-diagram is to attach.

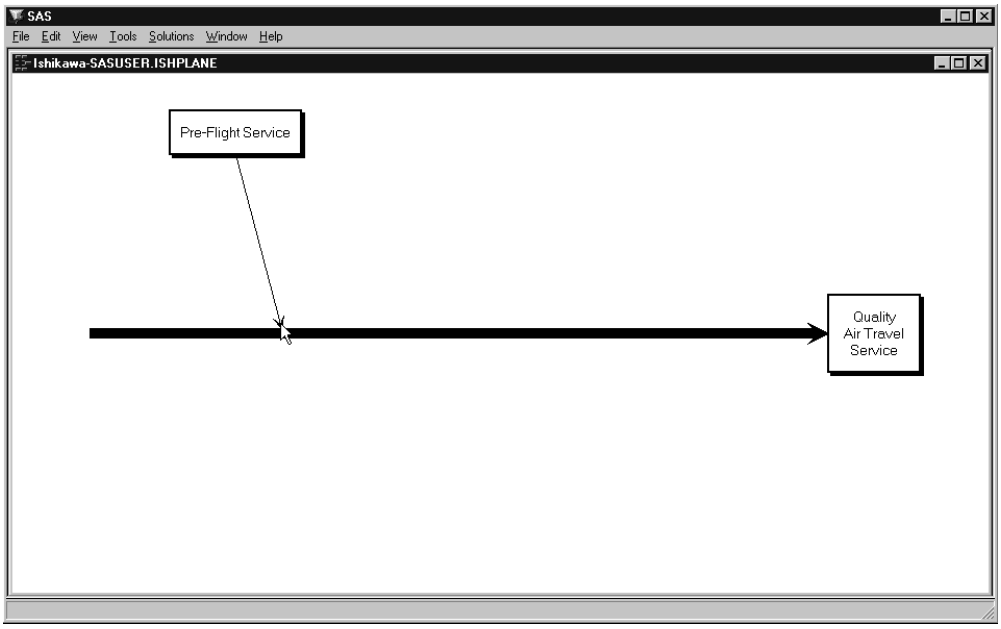


Figure 25.44. Constructing a Master Diagram

To complete the diagram, repeat the process for the remaining branches.

Creating Graphics Output Using SAS/GRAPH Software

One way to create a hard copy of your Ishikawa diagram is to send it to a graphics device using SAS/GRAPH software. To do this, you should submit a GOPTIONS statement to direct the graphics output to the appropriate location and control the output format *before you invoke the ISHIKAWA environment*. For example, the following GOPTIONS statement directs the output to a PostScript device:

```
goptions target=psl noprompt;
```

If you do not specify a target device before invoking the ISHIKAWA environment, you will be prompted for one before the graph is generated.

In the ISHIKAWA environment, when you are ready to route your output to a hard copy device, select **File** ▷ | **Save as** ▷ | **Graph**. This opens a dialog that enables you to customize various aspects of your graph.

Figure 25.45. Hard Copy Requestor

To save the diagram to the default graphics catalog in the WORK library (WORK.GSEG), simply press **OK** and close the dialog. The default member name is ISHIKAWA.

To save the diagram to a different graphics catalog, select **Save...** and then use the Member selector window to specify a library, a SAS catalog, and a member name.

When sending a diagram directly to an output device, you can ignore the member name entirely.

To save to your own graphics catalog, select **Save...** and then use the Save a member selection window to specify a catalog and data set name. Simply select **OK** when you want to save your diagram to the default graphics catalog (WORK.GSEG). When sending a diagram directly to an output device, you can use **OK**.

You must specify two SAS/GRAPH fonts for drawing the labels in the hard copy of the diagram. The hardware fonts used in the ISHIKAWA environment cannot be used for your hard copy. The *primary* font and size are used for the first three levels of text. The *secondary* font and size are used for the remaining levels of text.

To change fonts, enter a valid SAS/GRAPH font name in the font field or click on the button to the right of the font field to display a font requestor dialog. The default font is SIMPLEX.

You can specify the height of the text directly in the height field (in screen percent units), or you can click on the button to the right of the field to request an *absolute* height or a *relative* height.

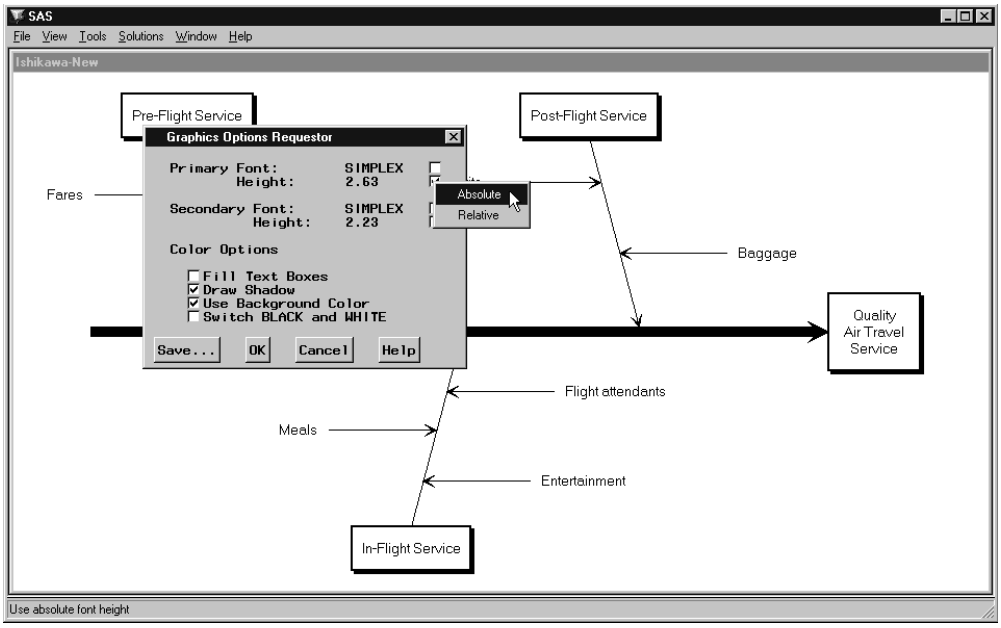


Figure 25.46. Font Height Selector

Select **Absolute** when you want the font height in the output to be the same height as the font height used in the ISHIKAWA environment even if the output window and the ISHIKAWA window differ in size. Select **Relative** to maintain the same font height to window size proportion in both the ISHIKAWA window and the output window. The numeric value entered in the height field after either choice is a screen percent unit. The default text height is *absolute*.

Use the ◇ **Fill Text Boxes** and ◇ **Draw Shadow** check boxes to suppress the box fills and box shadows from the output. They cannot be used to *add* these features to the hard copy if they were not present in the ISHIKAWA window.

Use the ◇ **Use background color** check box to indicate whether the background color from the ISHIKAWA environment is used in the output. This option is useful when you are sending your diagram to a *color* device and you want the background in your hard copy to match that of your ISHIKAWA environment.

Use the ◇ **Switch Black and White** check box to interchange black and white when the diagram is sent to the output device. This option is useful when you send your diagram from a white-on-black display to a black-on-white hard copy device.

Click on **OK** to generate the hard copy output or click on **Cancel** to quit.

Creating Bitmap Graphics Output

A second way to create a hard copy of your Ishikawa diagram is to export it as a bitmap to one of the following:

- the host graphical clipboard
- an external bitmap file
- a SAS/GRAPH Image catalog entry

To copy the Ishikawa diagram as a bitmap to the host clipboard, select **Edit** ▷ | **Copy** |. The results are host specific. For more details about copying to the host clipboard on your system, consult the SAS companion for your host.

To export the Ishikawa diagram to a bitmap file using SAS/GRAPH software, select **File** ▷ | **Export as Bitmap** ▷ | **File...** |.

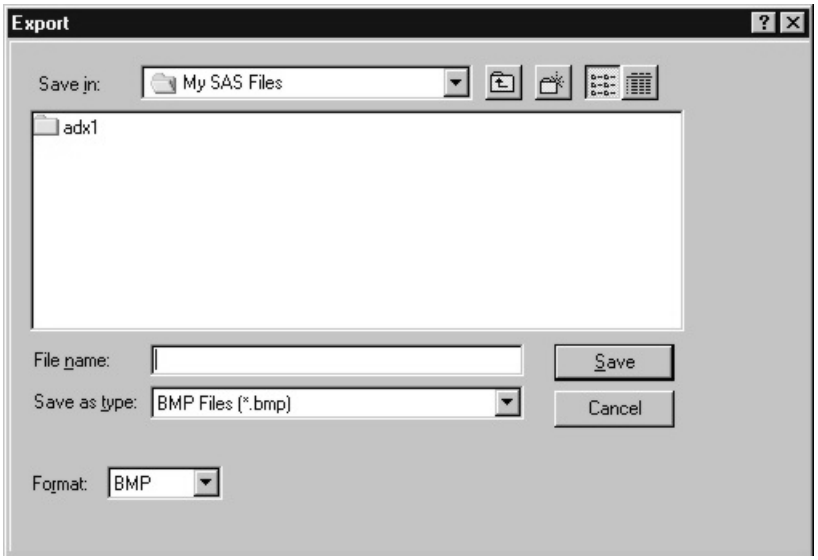


Figure 25.47. Export File Requestor

The appearance of this dialog will be host specific. For more details about the format of this dialog on your system, consult the SAS companion for your host.

To save the Ishikawa diagram as an IMAGE entry in a SAS catalog, select **File** ▷ | **Save as** ▷ | **Image** |.

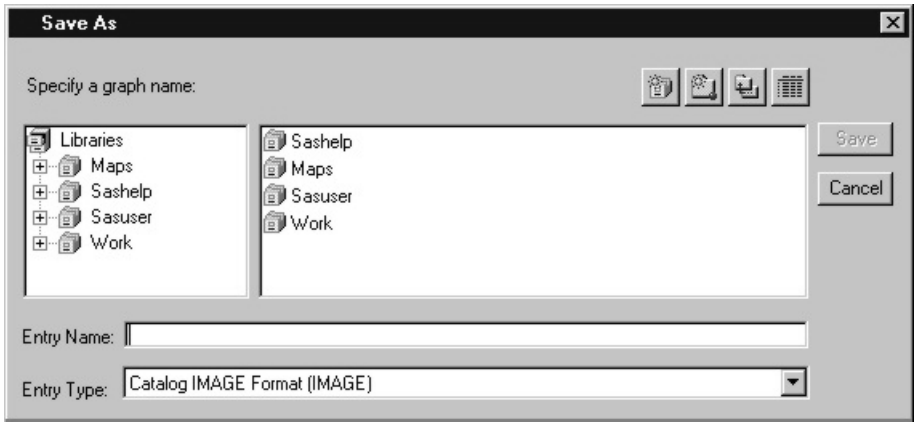


Figure 25.48. Entry Selector

You must specify a SAS catalog in which to save the IMAGE entry as well as a name for the object.

When exporting your diagram to a bitmap or saving to a SAS/GRAPH IMAGE entry, you can have the colors mapped so that color diagrams are saved in black on white or white on black. You do not have to make those changes to the diagram yourself. Use **File** ▷ | **Export as Bitmap** ▷ | **Customize...** to display the following dialog:

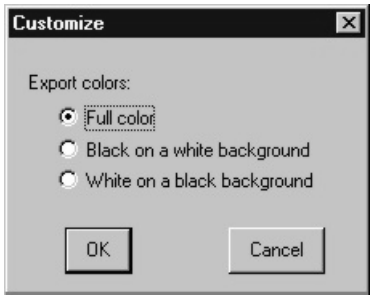


Figure 25.49. Customize Export Dialog

Select ◇ **Black on white** to convert the output to a black diagram on a white background. This is useful when the diagram is being exported to a document.

Select ◇ **White on black** to convert the output to a white diagram on a black background. This is useful when the diagram is being exported for display on a black and white terminal.

Modifying Fonts

By default, the ISHIKAWA environment uses the same hardware font as the SAS windowing environment. However, you have the option of specifying two different font styles/sizes.

The *primary* font is used for labeling arrows in the first three levels of the diagram. The *secondary* font is used for labeling arrows in the remaining levels. You will typically use a smaller font in the detailed (secondary) areas of the diagram.

To change a font, select **View ▷** | **Ishikawa Settings ▷** | **Primary Fonts...** or **View ▷** | **Ishikawa Settings ▷** | **Secondary Fonts...** to display the Font Requestor window, as follows:

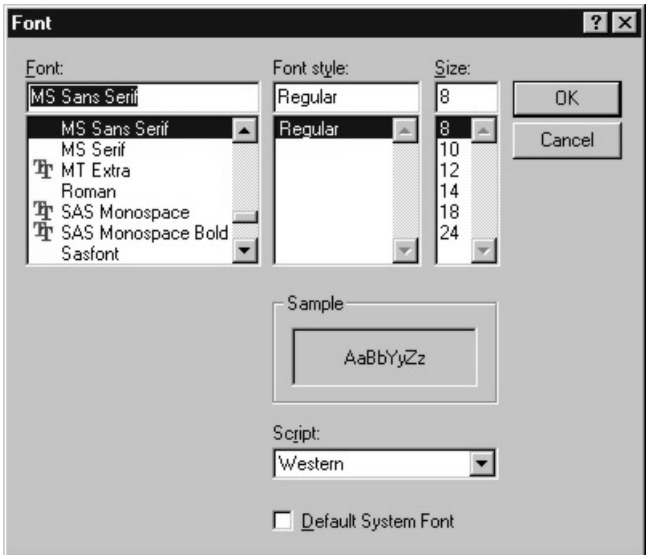


Figure 25.50. Font Requestor

The layout of the Font requestor window is host specific. Typically, it will contain a list of available fonts and sizes displayed in a scrollable region. Refer to your host documentation for specific information regarding the format of this dialog.

To change fonts, select a font from the list.

You must close the Font Requestor window before you can proceed. Select **OK** to apply the font or **Cancel** to cancel the dialog.

To customize your session so that these fonts are permanently associated with the ISHIKAWA environment, select **View ▷** | **Save Attributes** from the command bar.

Modifying Box Colors

By default, the box fill (background) color is empty and the shadow (outline) color is the same as the arrow color.

To modify the colors associated with trunk and branch boxes, select **View ▷** | **Ishikawa Settings ▷** | **Colors...**. A dialog, similar to the following, is displayed:

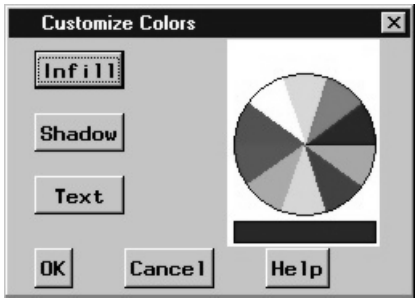


Figure 25.51. Colors Dialog

To change the fill color of all the boxes* in the Ishikawa diagram, do the following:

- Select a color from the color palette.
- Select **Infill**.

Once modified, the fill color is unaffected by changes in the arrow color. To return the box to an empty fill, proceed as follows:

- Select the current infill color from the color palette (if it is not already the current color).
- Select **Infill**.

To change the shadow color of the boxes, select **Shadow** and follow the same procedure.

Select **OK** to close the dialog or **Cancel** to cancel the changes.

Modifying Arrow Colors and Line Styles

The ISHIKAWA environment provides a line style palette and a color palette that you can use to customize the arrows in your Ishikawa diagram. Select **View** ▷ **Palettes** to activate both palettes.†

* You cannot directly modify the colors of individual boxes from this dialog.
† If you are working on a black-and-white terminal, you should not use the color palette.

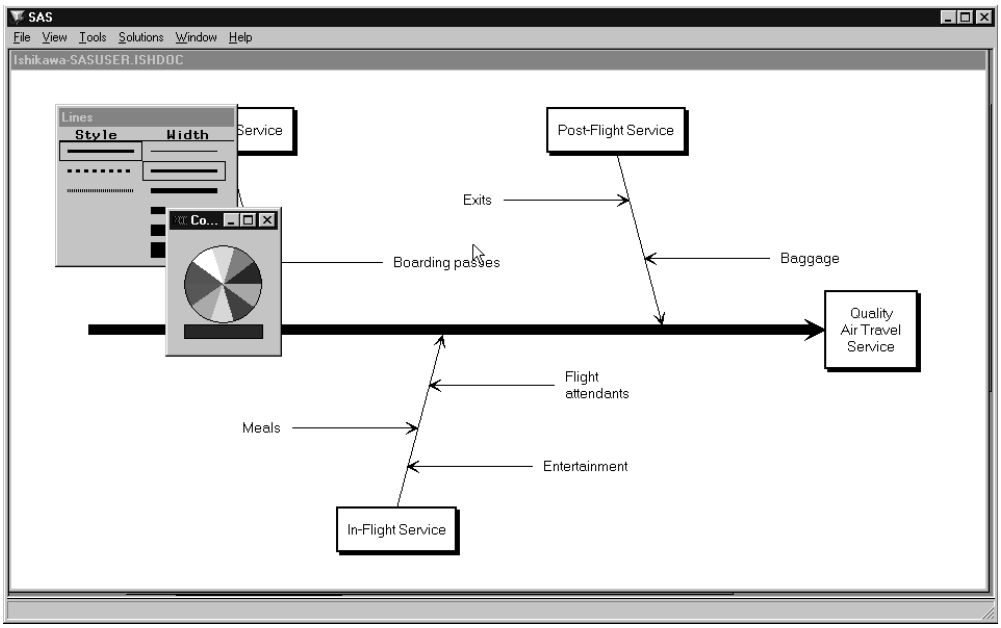


Figure 25.52. Line Style and Color Palettes

To specify the arrows to which color and line selections apply, subset them with the subset function. To toggle an arrow in or out of the list of subsetted arrows, do the following:

- Use the right mouse button to display the arrow head or the arrow tail popup menu. To subset an arrow and all its descendants, use the arrow head popup menu. Use the arrow tail popup menu to subset an arrow without any descendants.
- Select **Subset**.

The labels of all subsetted arrows are underlined.

On some hosts, shift-clicking on the arrow head or tail will also subset the arrow. You can subset any combination of arrows in the diagram.

You can change the color of all the subsetted arrows by selecting the desired color in the color palette with the mouse. Likewise, use the line palette to control the style and width of the arrows.

To unsubset all the arrows in the diagram, do the following:

- Move the cursor to a *background* area of the ISHIKAWA window.
- Use the right mouse button to activate the background popup menu.
- Select **Unsubset** from the popup menu.

To unsubset a specific arrow in the diagram, select **Subset** from the context-sensitive popup menu for the arrow head or tail.

Be sure to remove all subsets once you have finished modifying the diagram, since subsets affect the focus of many other operations.

Example

Continuing with the diagram from the previous section, subset the trunk using the arrow tail popup menu.

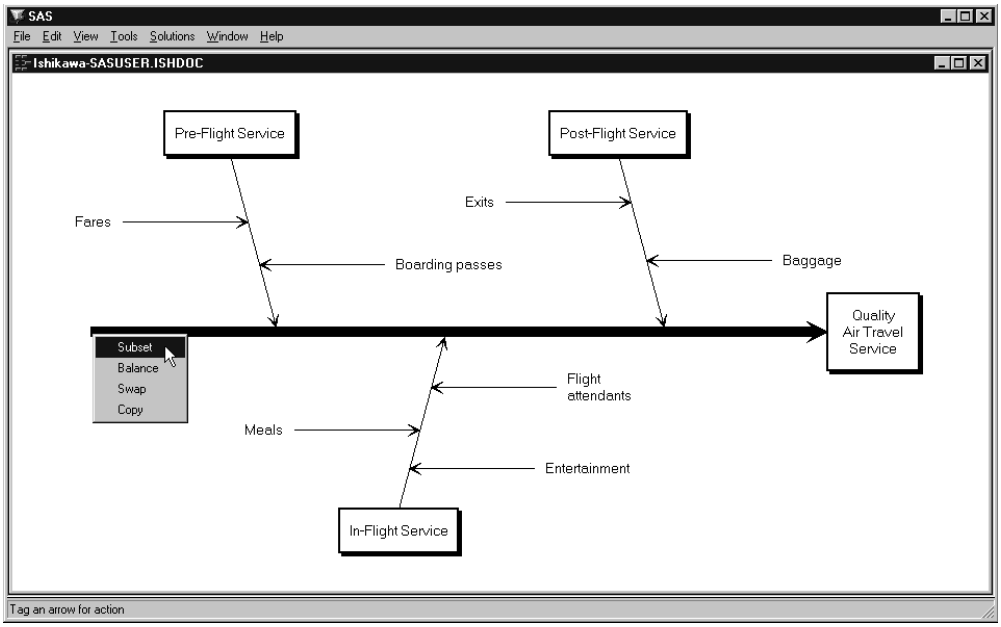


Figure 25.53. Subsetting Only the Trunk

Note that only the trunk is subsetted (as indicated by the underlined label).

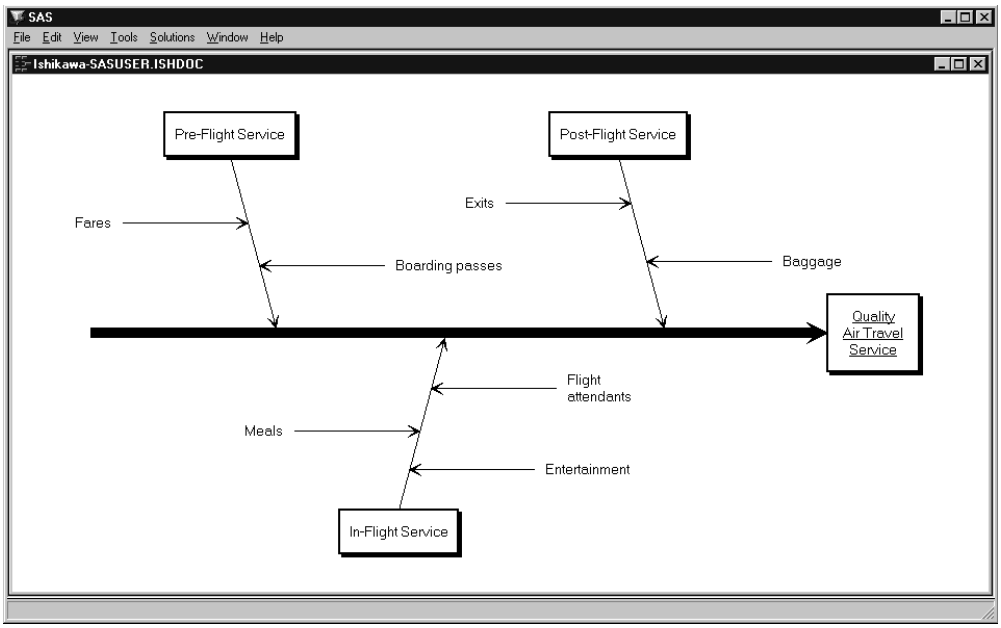


Figure 25.54. Subsetting Only the Trunk *(continued)*

When you select a line style from the line palette, only the line style of the subsetted arrow is changed.

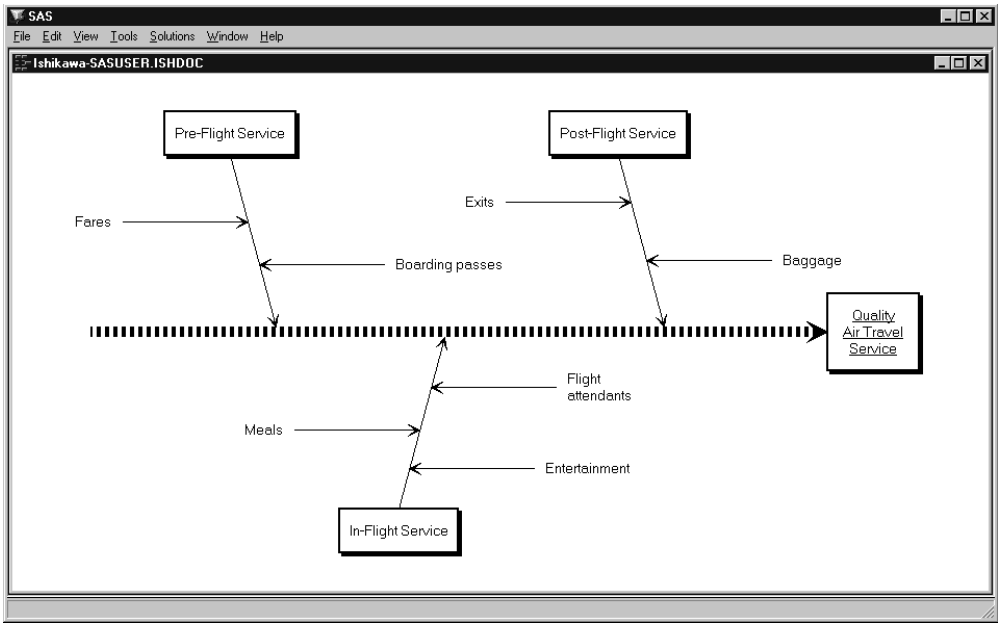


Figure 25.55. Modified Diagram

Alternately, if you subset the trunk using the arrow head popup menu, all of the arrows in the diagram are subsetted.

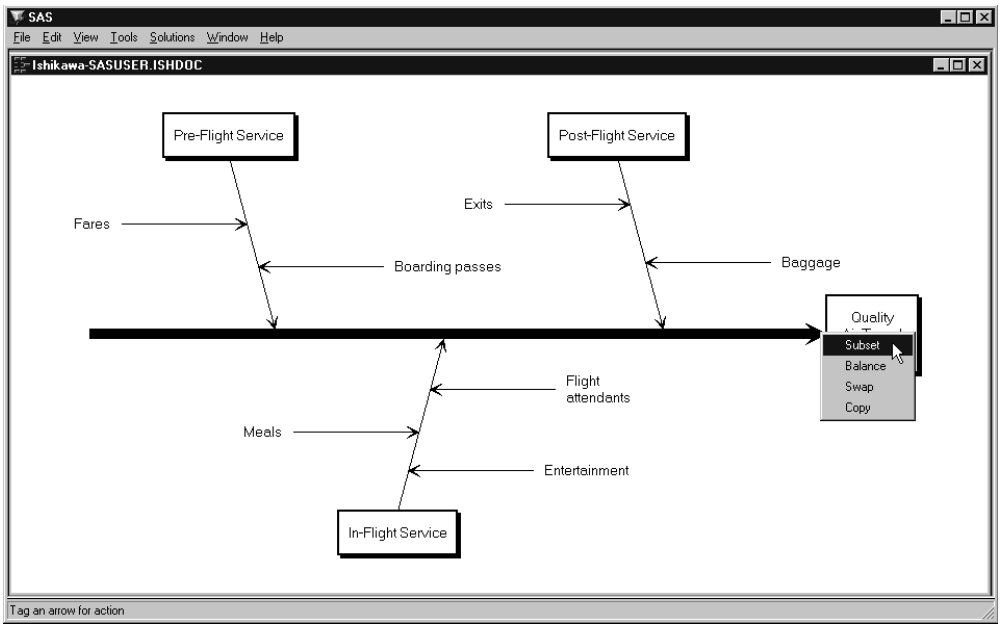


Figure 25.56. Subsetting the Entire Diagram

Note that all of the labels in the diagram are now underlined.

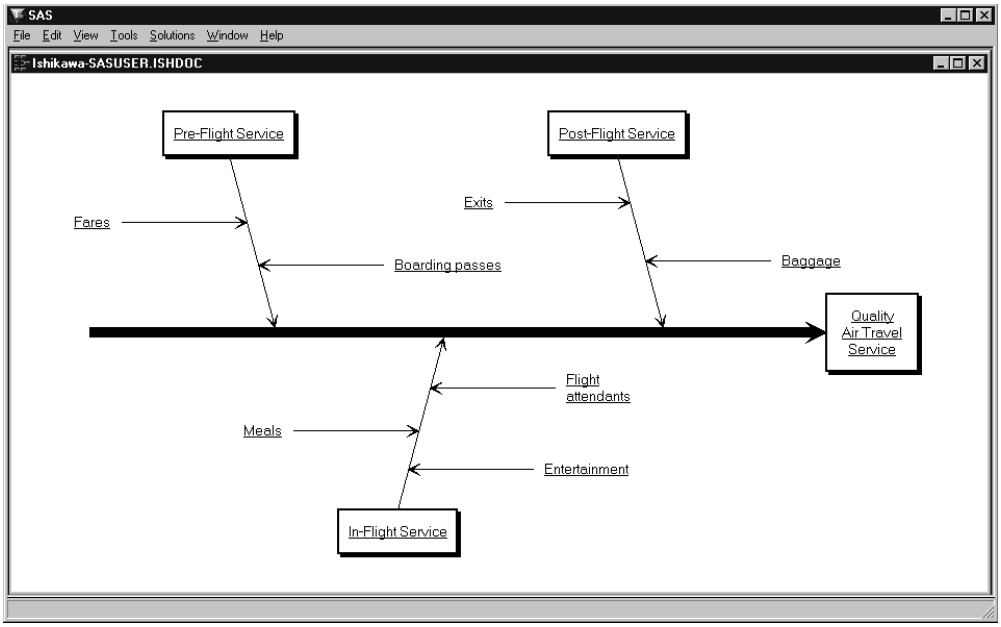


Figure 25.57. Subsetting the Entire Diagram *(continued)*

Now, when you select a new line style from the line palette, all the arrows are drawn with this line style.

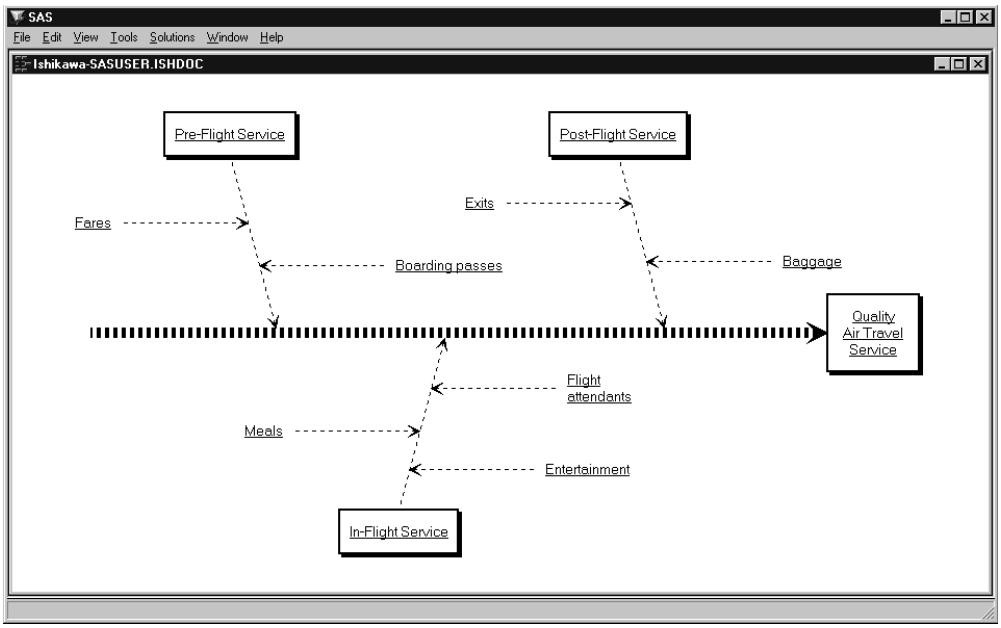


Figure 25.58. Modified Diagram

To remove the subset from the *Pre-Flight Service* branch and all its descendants, select **Subset** from the arrow head popup menu.

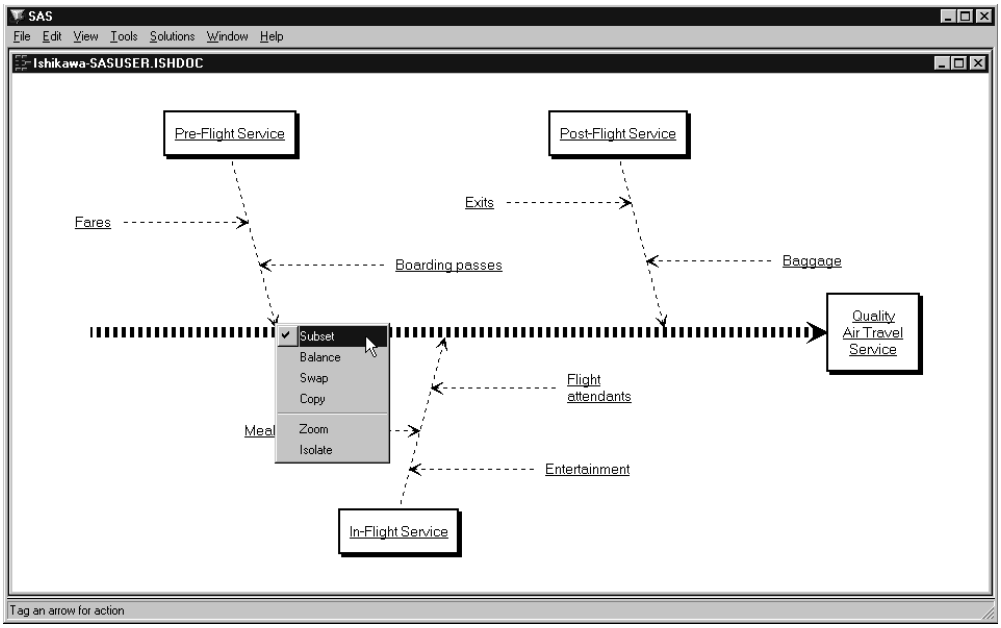


Figure 25.59. Selectively Removing Tags

This removes the underlines from the labels in these arrows.

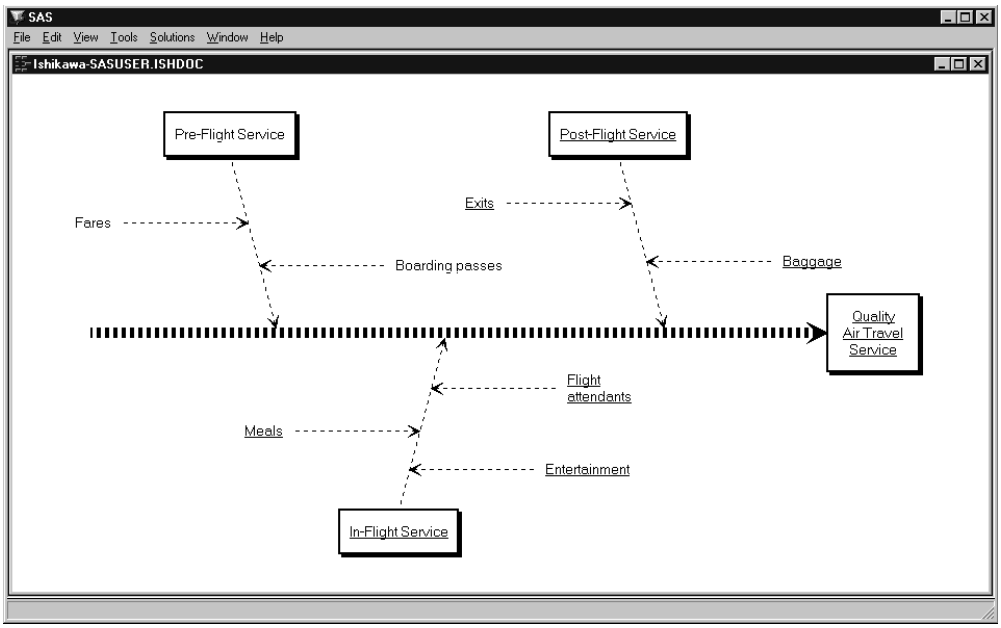


Figure 25.60. Selectively Removing Subsets *(continued)*

You can now use the line palette to change the line style for all the arrows in the diagram with the exception of the *Pre-Flight Service* branch and its descendants:

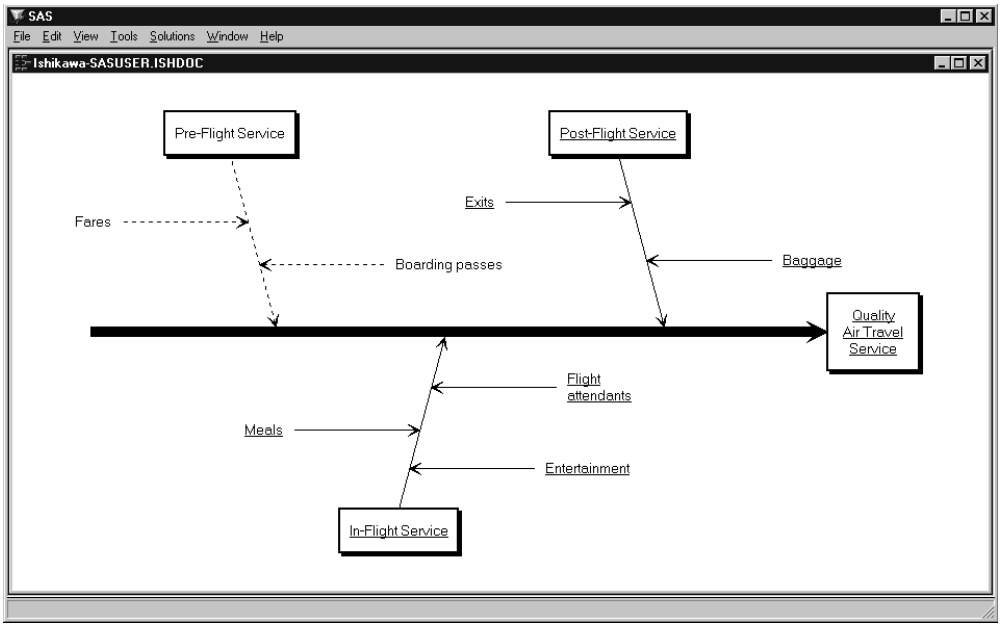


Figure 25.61. Modified Diagram

The same principles apply when making color changes—simply use the color palette instead of the line style palette.

Modifying Text Colors

By default, labels have the same color as the arrow. To modify the text color independently of the arrow color, select **View ▷** | **Ishhikawa Settings ▷** | **Colors...** . The Customize Color window, similar to the following, will open:

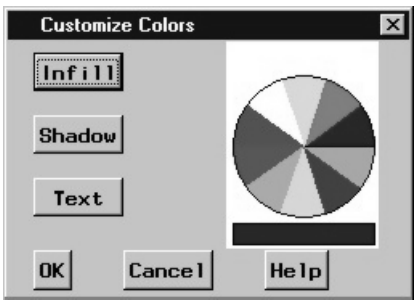


Figure 25.62. Colors Dialog

To change the text color of all the arrows* in the Ishikawa diagram, do the following:

- Select a color from the color palette.
- Select **Text** .

Once modified, the text color is unaffected by changes to the arrow color. To relink the text color to the arrow color, do the following:

- Select the current text color from the color palette (if it is not already the current color).
- Select **Text** .

Select **OK** to close the dialog window. To cancel the changes, select **Cancel** .

Modifying Arrow Heads

To modify the characteristics of the arrow heads in your diagram, select **View ▷** | **Ishikawa Settings ▷** | **Arrows...** . This opens the following dialog:

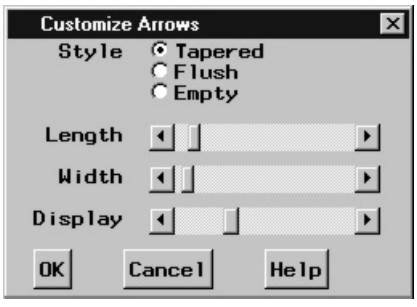


Figure 25.63. Arrows Dialog

* You cannot directly modify the text color for individual arrows from this dialog.

The dialog controls the characteristics of all arrow heads. Arrow heads cannot be modified individually.

Arrow heads can be tapered, flush, or empty. Use the sliders labeled **Length** and **Width** to control the length and width of the arrow heads. Move the sliders to the right to increase the length/width of the arrow head and to the left to decrease the length/width.

Removing arrow heads increases the readability of a highly detailed diagram. Use the **Display** slider to control the level at which arrow heads are displayed. Move the slider to the extreme left to remove all the arrow heads and to the extreme right to display all the arrow heads. Use the intermediate settings to select a threshold level of detail, above which arrow heads are not displayed. By default, arrow heads are displayed for all levels.

Select **OK** to close the window. To cancel the changes, select **Cancel**.

Modifying Environmental Attributes

You can modify other features of the ISHIKAWA environment such as zooming, mouse sensitivity, and shadow attributes by selecting **View** ▷ | **Ishikawa Settings** ▷ | **Other...** to open the following dialog:

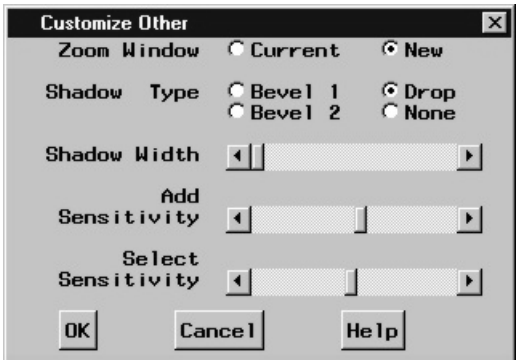


Figure 25.64. Others Dialog

Zoom Window controls whether the zoom operation opens a new window or draws in the current window. Select ◇ **Current** to reduce the amount of window management required.

The **Shadow Type** button controls the type of shadow that is drawn around the trunk and branch boxes.

- ◇ **Bevel 1** draws a beveled edge box with a lower-right light source.
- ◇ **Bevel 2** draws a beveled edge box with an upper-left light source.
- ◇ **Drop** draws a box with a drop shadow. This is the default.
- ◇ **None** suppresses the shadow.

The **Shadow Width** slider controls the shadow width if the boxes have shadows or the outline width when boxes are displayed without shadows. Move the slider to the right to increase the shadow width and to the left to decrease the width.

The **Add Sensitivity** slider controls how closely you must position the cursor to an existing arrow before a mouse click results in an add arrow operation. Move the slider to the right to increase the size of the context-sensitive area and to the left to reduce the size of the context-sensitive area.

The **Select Sensitivity** slider controls how closely you must position the cursor to an existing arrow before a mouse click results in an edit, delete, move, or popup arrow operation. Operate this slider in the same manner as the **Add Sensitivity** slider.

Saving an Ishikawa Diagram for Future Editing

You must save your Ishikawa diagram as a SAS data set if you intend to edit it in the future with the ISHIKAWA environment. The ISHIKAWA environment does not reconstruct Ishikawa diagrams by reading graphics entries (GRSEG) from SAS catalogs.

Select **File** ▷ | **Save As** ▷ | **Data Set** to activate the Data Set Requestor window.

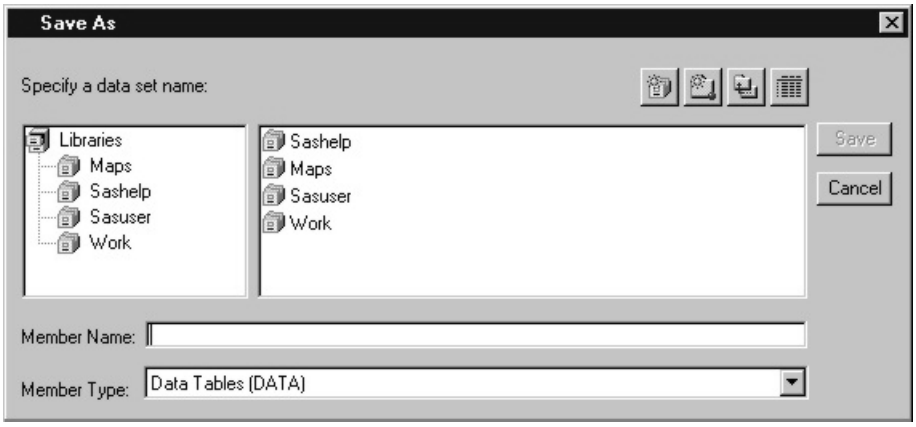


Figure 25.65. Output File Requestor

A list of SAS *librefs* is displayed in the Libraries tree in the left region of the dialog. Begin by selecting a libref from the list. A libref refers to a permanent SAS data library located on your host system. For example, the default SASUSER libref (on most hosts) points to a directory called SASUSER, located under the working directory of your current SAS session. Any data sets saved with the libref **SASUSER** will be saved in that directory.

To direct your SAS data sets to a different directory, select the *Create new library* tool icon to open the New Library dialog. Use this dialog to specify the directory and assign a libref to that directory.

To select the libref **SASUSER**, move your cursor over that entry in the list and click. The region to the right of the Libraries tree is used to display any existing SAS data sets in that library.

To save your diagram in an existing SAS data set, use the mouse to click on an entry in the list. The *member name* field will be updated to reflect your choice. If you want to save your diagram in a new SAS data set, move your cursor to the *member name* field and type the new name (in this example, SERVICE).

Select **Save** to save the diagram and return to the ISHIKAWA environment or select **Cancel** to cancel the save.

Reading an Existing Ishikawa Diagram

To enter the ISHIKAWA environment and resume editing an existing diagram, you must have previously saved the diagram as a SAS data set. The ISHIKAWA environment *does not* allow you to modify graphs stored in SAS/GRAPH catalogs.

You can specify the name of this data set when you establish the ISHIKAWA environment with the following statements:

```
proc ishikawa data=libref.dataset;
   run;
```

Alternatively, the ISHIKAWA environment will prompt you for a data set after you invoke the environment with the following statements:

```
proc ishikawa;
   run;
```

When you specify a data set in the PROC statement, the ISHIKAWA environment is initialized and your diagram is displayed up to the branch level. The message area will indicate if any additional detail is hidden. You can edit your diagram even if some of the diagram is hidden. To add or remove detail one level at a time, select **> Detail** or **< Detail** from the background popup menu.

When you do not specify a data set in the PROC statement, you will see the following menu:

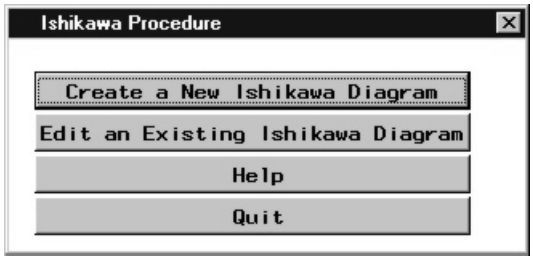


Figure 25.66. Initial Menu

Since you are editing an existing diagram rather than starting a new diagram, select **Edit an Existing Diagram** to activate the Member Selector window.

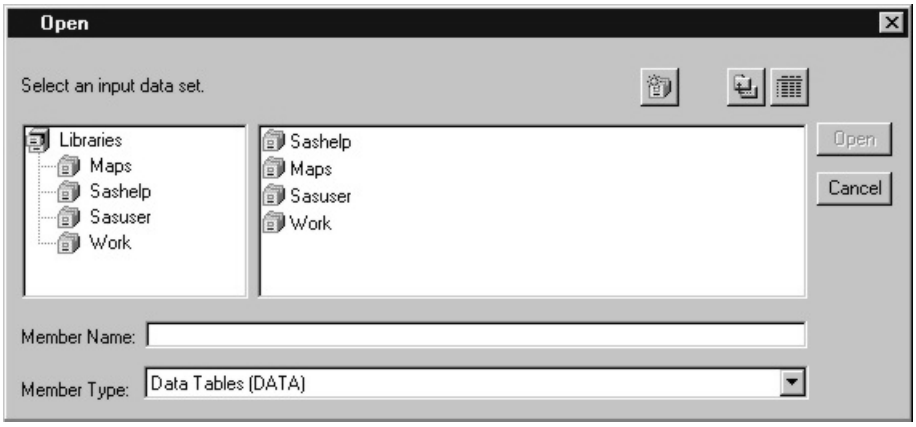


Figure 25.67. Input Member Selector

Use the Member Selector window to specify an input SAS data set. For information on how to specify the SAS data set name, follow the steps outlined in "Saving an Ishikawa Diagram for Future Editing" on page 738.

To establish the ISHIKAWA environment and display the diagram you have selected, select **Open**. The diagram is displayed up to the branch level.

To quit or start a new diagram, return to the main menu by selecting **Cancel**.

Displaying Multiple Ishikawa Diagrams

The ISHIKAWA environment enables you to view multiple Ishikawa diagrams simultaneously for side-by-side comparisons of different diagrams. You can also use this feature to transfer information between diagrams, since the move and copy operations function across windows. You can have up to four ISHIKAWA windows open at one time.

To open a window for another Ishikawa diagram, select **File** ▷ **Open**. This will display the Member Selector window, which you can use to specify the name of the input SAS data set for the other Ishikawa diagram.

You can also start new diagrams while displaying other Ishikawa diagrams. To open a window for a new Ishikawa diagram, select **File** ▷ **New**. This opens an ISHIKAWA window with a template for a new diagram.

Example

The following figure shows an Ishikawa diagram for *Quality Air Travel Service* after an initial brainstorming session:

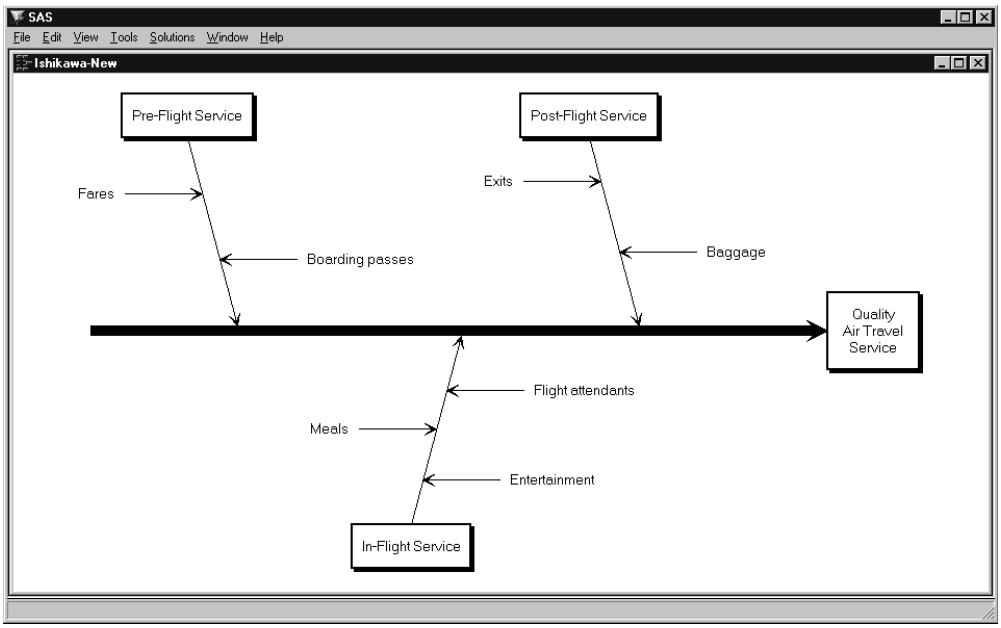


Figure 25.68. Single Ishikawa Diagram

The current diagram and another Ishikawa diagram can be viewed simultaneously by selecting **File** ▷ **Open** from the command bar.

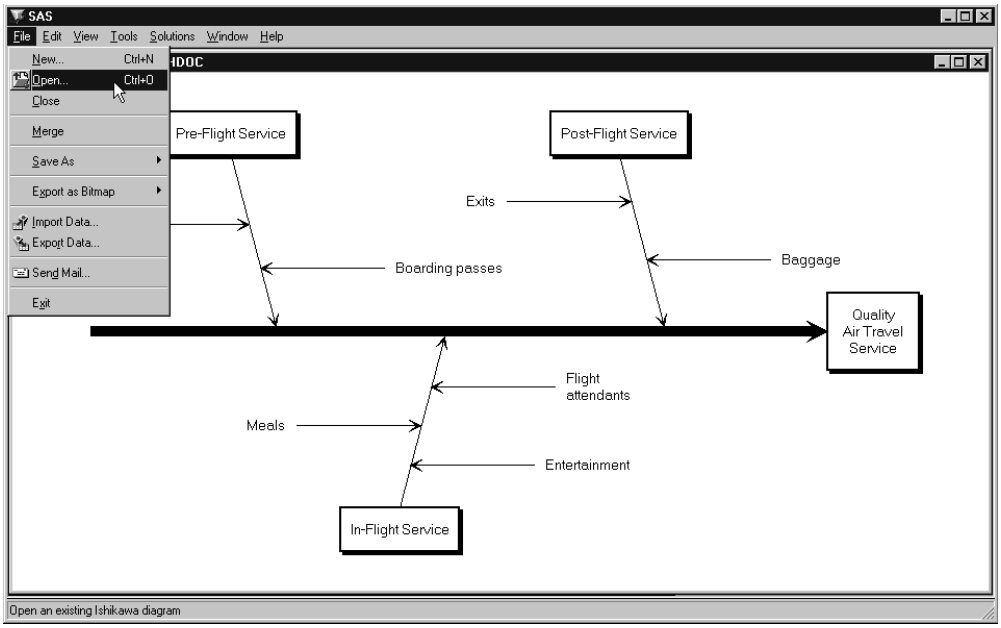


Figure 25.69. Opening a Second Diagram

In this situation, displaying both diagrams concurrently emphasizes the improved understanding of the process. It also enables you to transfer information from one

diagram to another.

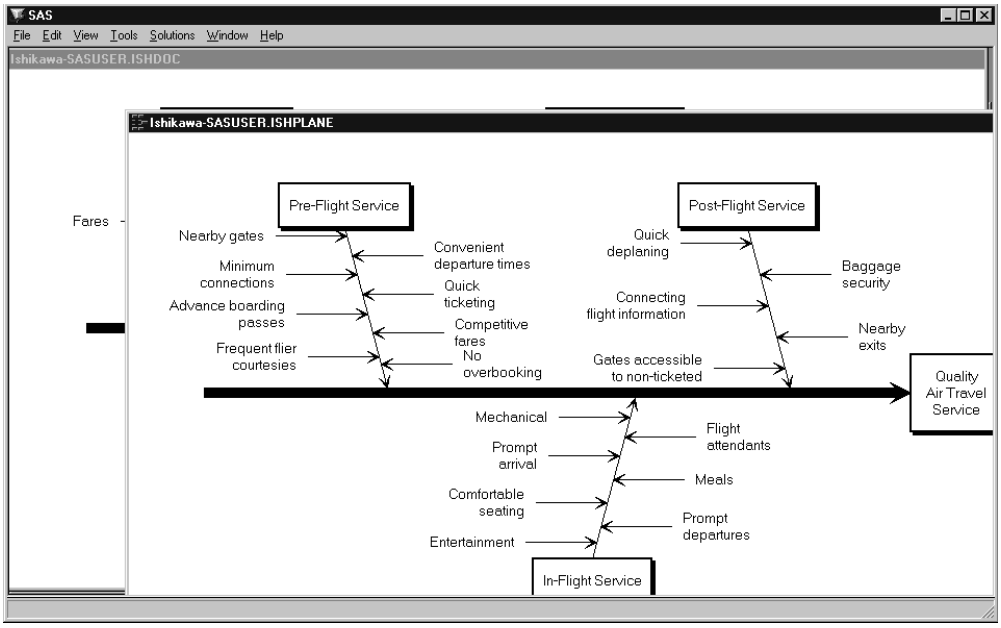


Figure 25.70. Viewing Multiple Ishikawa Diagrams

Input and Output Data Sets

ISHIKAWA Procedure

The following is a complete list of the variables in output SAS data sets created by the ISHIKAWA environment:

Variable	Type	Len	Description
LEVEL	Num	8	Level of detail
TEXT1	Char	40	First line of label
TEXT2	Char	40	Second line of label
TEXT3	Char	40	Third line of label
TEXT4	Char	40	Fourth line of label
TEXT5	Char	40	Fifth line of label
NOTE1	Char	40	First line of note
NOTE2	Char	40	Second line of note
NOTE3	Char	40	Third line of note
NOTE4	Char	40	Fourth line of note
RELPOS	Num	8	Relative arrow position
SIDE	Char	1	Side arrow attaches to parent
ANGLE	Num	8	Angle (non-horizontal arrows)
LWIDTH	Num	8	Line width
LSTYLE	Num	8	Line style
LCOLOR	Char	8	Line color

TCOLOR	Char	8	Text color
ICOLOR	Char	8	Box infill color
SCOLOR	Char	8	Shadow color
STYPE	Char	1	Shadow type
SWIDTH	Num	8	Shadow width
RELLNG	Num	8	Relative length of an arrow
HLEVEL	Num	8	Arrow head threshold
HSTYLE	Num	8	Arrow head style
HLNGTH	Num	8	Arrow head length
HWIDTH	Num	8	Arrow head width
HTEXT	Num	8	Font height
FTEXT	Char	8	Font

Only the variables _LEVEL_ and _TEXT1_ are required in the input data set for the ISHIKAWA procedure. Each observation in the input data set corresponds to a particular arrow in the diagram. The order of the observations is critical because it defines the relationships of the arrows.

- The trunk is always the first observation.
- The remaining observations are ordered so that leaves are nested within stems, stems are nested within branches, and branches are nested within the trunk.
- The variable _LEVEL_ is numeric and indicates the level within the diagram. The trunk has a level of 0, branches have a level of 1, stems have a level of 2, and so on.
- The first line of text in a label is stored as _TEXT1_, the second as _TEXT2_, and so on.

Example

The following is a partial listing of the SAS data set used to create the Ishikawa diagram shown in Figure 25.4 on page 694:

Obs	_level_	_text1_	_text2_	_text3_
1	0	Quality	Air Travel	Service
2	1	Pre-Flight Service		
3	2	Competitive	fares	
4	2	Convenient	departure times	
5	2	Quick	ticketing	
6	2	Frequent flier	courtesies	
7	1	In-Flight Service		
8	2	Prompt	departures	
9	2	Comfortable	seating	

Figure 25.71. Input SAS Data Set

Note the structure of this data set:

- The trunk (always the first observation) has a _LEVEL_ value of zero.

- All subsequent observations for which _LEVEL_ is equal to one are branches that emerge from the trunk.
- Observations 4 and 5 are both leaves that emerge from the preceding stem (observation 3).
- Likewise, leaves 7 and 8 emerge from the preceding stem (observation 6).

You can use this data set as a way of extracting text and notepad information from the diagram.

Syntax

There are only three options that can be specified in the PROC ISHIKAWA statement, since the ISHIKAWA procedure is primarily a user-driven procedure.

DATA=*SAS-data-set*
identifies the name of a SAS data set that specifies an existing Ishikawa diagram. By default, the procedure will prompt you to edit an existing Ishikawa diagram or start a new one. When you specify the DATA= option, the procedure bypasses this initial menu. For example, the following statements simplify editing an existing Ishikawa diagram saved in a SAS data set:

```
proc ishikawa data=work.airline;
   run;
```

NEW
starts a new Ishikawa diagram. By default, the procedure will prompt you to edit an existing Ishikawa diagram or start a new one. When you specify the NEW option, the procedure bypasses this initial menu and starts with a new diagram. Do not specify any other options when using the NEW option. For example, the following statements simplify starting a new Ishikawa diagram:

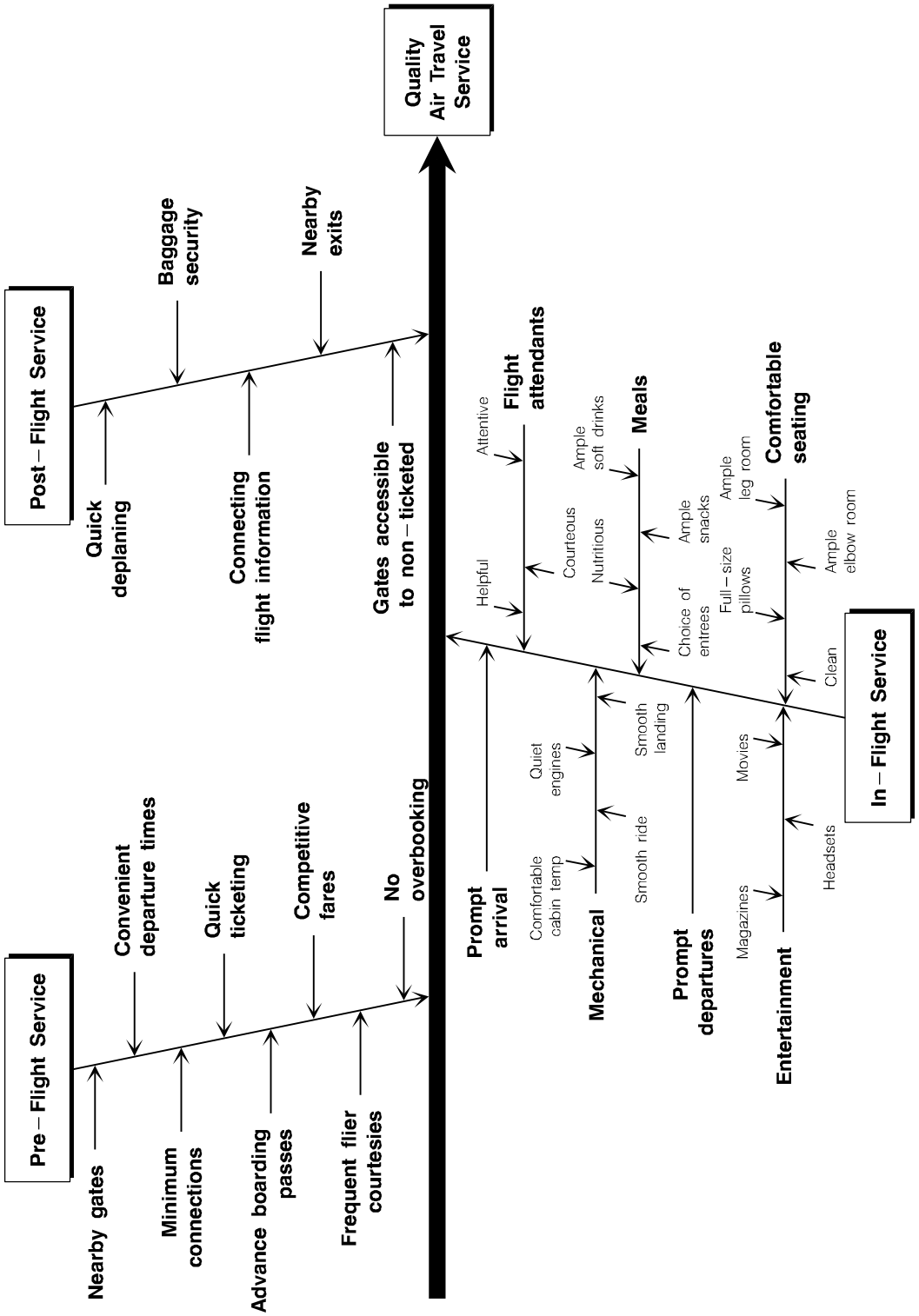
```
proc ishikawa new;
   run;
```

NOFS
allows you to create hard copies of Ishikawa diagrams saved as SAS data sets without invoking the interactive features of the procedure. You must specify the DATA= option when you use the NOFS option. For example, the following statements create a hard copy of the Ishikawa diagram saved in the SAS data set *work.airline*:

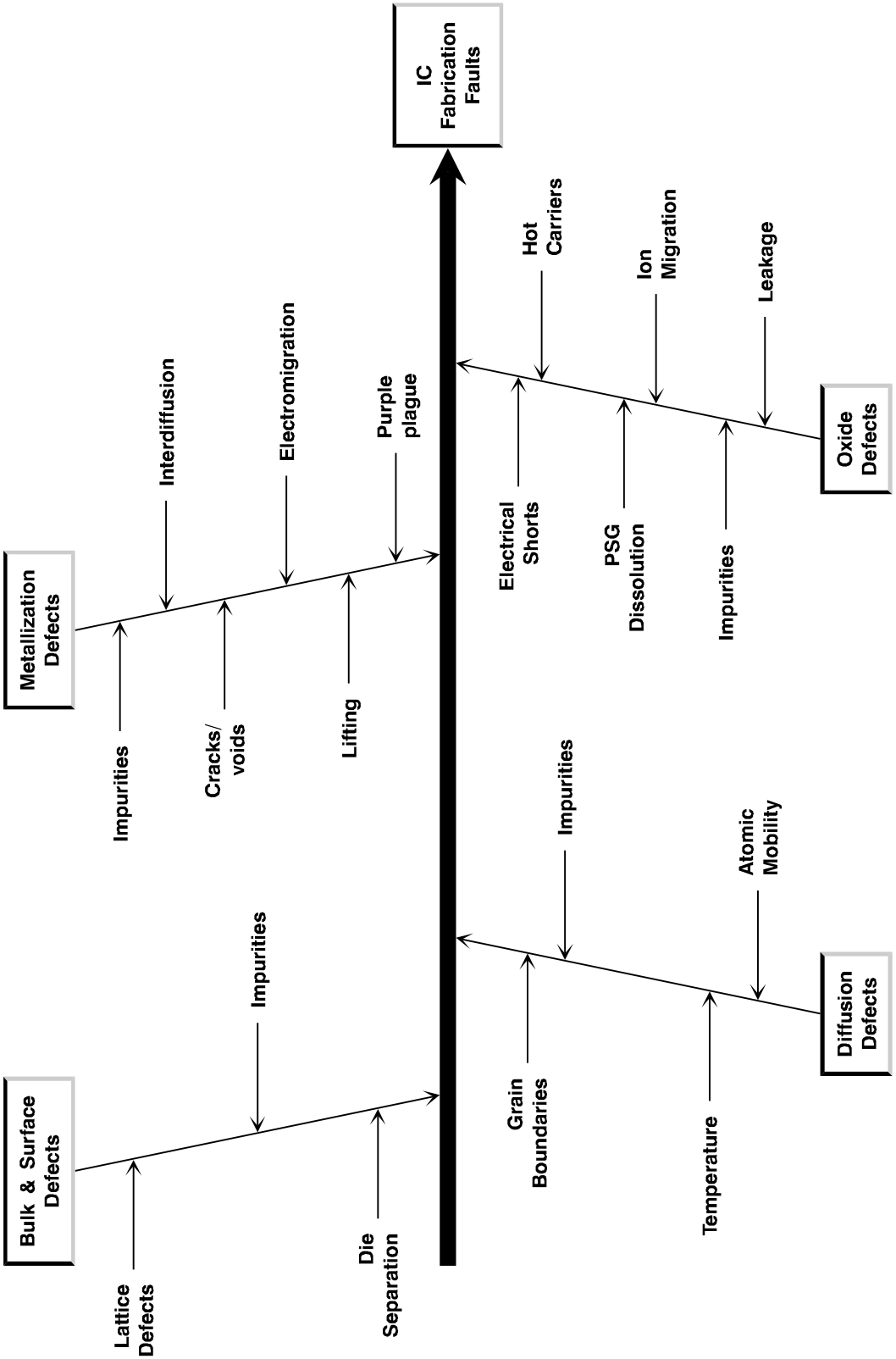
```
goptions dev=psl noprompt;
proc ishikawa data=work.airline nofs;
   run;
```

Examples

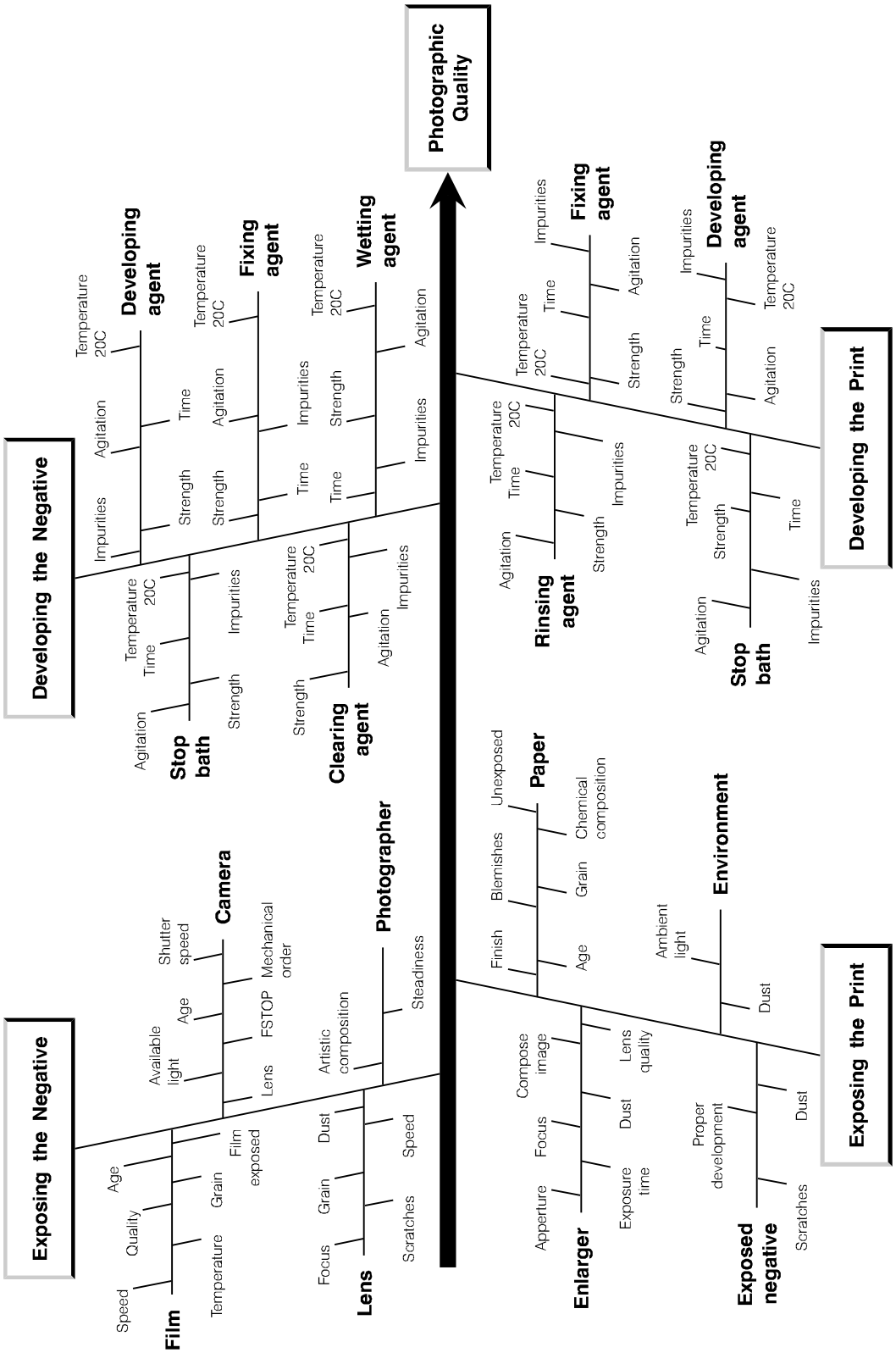
Example 25.1. Quality of Air Travel Service



Example 25.2. Integrated Circuit Failures



Example 25.3. Photographic Development Process



References

Ishikawa, K. (1982), *Guide to Quality Control (Second Revised English Edition),* Tokyo: Asian Productivity Organization.

Karabatsos, N. A. (1989), "In Memoriam: Dr. Kaoru Ishikawa: Quality Organizer," *Quality Progress*, 22, No. 6, 20.

Kume, H. (1985), *Statistical Methods for Quality Improvement,* Tokyo: AOTS Chosakai, Ltd.

Rodriguez, R. N. (1991), "Applications of Computer Graphics to Two Basic Statistical Quality Improvement Methods," *Proceedings of the National Computer Graphics Association Conference,* Chicago, April 1991, 17–26.

Sarazen, J. S. (1990), "The Tools of Quality, Part II: Cause-and-Effect Diagrams," *Quality Progress*, 23, No. 7, 59–62.

SAS Institute Inc. (1999), *SAS Language Reference: Dictionary, Version 8,* Cary, NC: SAS Institute Inc.

Part 6
The MACONTROL Procedure

Contents

Introduction 753

Chapter 26. PROC MACONTROL Statement 755

Chapter 27. EWMACHART Statement 763

Chapter 28. MACHART Statement 819

Chapter 29. INSET Statement 863

References 869

Introduction

The MACONTROL procedure creates moving average control charts, which are tools for deciding whether a process is in a state of statistical control and for detecting shifts in a process average. The procedure creates the following two types of charts:

- *uniformly weighted moving average charts* (commonly referred to as *moving average charts*). Each point on a moving average chart represents the average of the w most recent subgroup means, including the present subgroup mean. The next moving average is computed by dropping the oldest of the previous w subgroup means and including the newest subgroup mean.

  The constant w, often referred to as the *span* of the moving average, is a parameter of the moving average chart. There is an inverse relationship between w and the magnitude of the shift to be detected; larger values of w are used to guard against smaller shifts.

- *exponentially weighted moving average (EWMA) charts*, also referred to as *geometric moving average (GMA) charts*. Each point on an EWMA chart represents the weighted average of all the previous subgroup means, including the mean of the present subgroup sample. The weights decrease exponentially going backward in time.

  The weight r $(0 < r \leq 1)$ assigned to the present subgroup sample mean is a parameter of the EWMA chart. Small values of r are used to guard against small shifts. If $r = 1$, the EWMA chart reduces to a Shewhart $\overline{X}$ chart.

In the MACONTROL procedure, the EWMACHART statement produces EWMA charts, and the MACHART statement produces uniformly weighted moving average charts.

In contrast to the Shewhart chart where each point is based on information from a single subgroup sample, each point on a moving average chart combines information from the current sample and past samples. Consequently, the moving average chart is more sensitive to small shifts in the process average. On the other hand, it is more difficult to interpret patterns of points on a moving average chart, since consecutive moving averages can be highly correlated, as pointed out by Nelson (1983).

You can use the MACONTROL procedure to

- read raw data (actual measurements) or summarized data (subgroup means and standard deviations) to create charts
- specify control limits as probability limits or in terms of a multiple of the standard error of the moving average
- adjust the control limits to compensate for unequal subgroup sample sizes

- accept numeric- or character-valued subgroup variables
- display subgroups with date and time formats
- estimate the process standard deviation σ using a variety of methods or specify a standard (known) value for σ
- analyze multiple process variables in the same chart statement
- provide multiple chart statements. If used with a BY statement, the procedure generates charts separately for BY groups of observations.
- tabulate the information displayed in the control chart
- save moving averages, control limits, and control limit parameters in output data sets
- superimpose plotted points with stars (polygons) whose vertices indicate the values of multivariate data related to the process
- display a trend chart below the moving average chart that plots a systematic or fitted trend in the data
- produce charts on line printers or on graphics devices. Charts produced on line printers can use special formatting characters that improve the appearance of the chart. Charts produced on graphics devices can be annotated, saved, and replayed.

Learning about the MACONTROL Procedure

If you are using the MACONTROL procedure for the first time, begin by reading Chapter 26, "PROC MACONTROL Statement," to learn about input data sets. Then turn to the "Getting Started" section of Chapter 27, "EWMACHART Statement," on page 766 or the "Getting Started" section of Chapter 28, "MACHART Statement," on page 822. These chapters also provide syntax information, computational details, and advanced examples.

Chapter 26
PROC MACONTROL Statement

Chapter Contents

OVERVIEW 757

SYNTAX 758
Input and Output Data Sets 761

Chapter 26
PROC MACONTROL Statement

Overview

The PROC MACONTROL statement starts the MACONTROL procedure and it identifies input data sets.

After the PROC MACONTROL statement, you provide either an EWMACHART or an MACHART statement that specifies the type of moving average chart you want to create and the variables in the input data set that you want to analyze. For example, the following statements request a uniformly weighted moving average chart:

```
proc macontrol data=values;
   machart weight*lot / mu0    = 8.10
                        sigma0 = 0.05
                        span   = 5;
run;
```

In this example, the DATA= option specifies an input data set named VALUES that contains the *process* measurement variable WEIGHT and the *subgroup-variable* LOT. *

You can use options in the PROC MACONTROL statement to

- specify input data sets containing variables to be analyzed, parameters for calculating moving averages, or annotation information
- specify a graphics catalog for saving graphical output
- specify that charts are to be produced on graphics devices or line printers
- define characters used for features on charts produced on line printers

In addition to the chart statement, you can provide BY statements, ID statements, TITLE statements, and FOOTNOTE statements. If you are using a graphics device, you can also provide graphics enhancement statements, such as SYMBOLn statements, which are described in *SAS/GRAPH Software: Reference*.

Note: If you are using the MACONTROL procedure for the first time, you should also read the "Getting Started" section on page 766 of Chapter 27, "EWMACHART Statement," and the "Getting Started" section on page 822 of Chapter 28, "MACHART Statement."

*In Release 6.12 and previous releases of SAS/QC software, the keyword GRAPHICS was required in the PROC MACONTROL statement to specify that the chart be created with a graphics device. In Version 7, you can specify the LINEPRINTER option to request line printer plots.

Syntax

The syntax for the PROC MACONTROL statement is as follows:

PROC MACONTROL < *options* >**;**

The PROC MACONTROL statement starts the MACONTROL procedure, and it optionally identifies various data sets and requests graphics output. You can specify the following *options* in the PROC MACONTROL statement. The marginal notes *Graphics* and *Line Printer* identify options that apply to graphics devices and line printers, respectively.

ANNOTATE=*SAS-data-set*
ANNO=*SAS-data-set*

Graphics

specifies an input data set that contains appropriate annotate variables, as described in *SAS/GRAPH Software: Reference*. The ANNOTATE= option allows you to add features to the moving average chart (for example, labels that explain out-of-control points). The ANNOTATE= data set is used only when the chart is created using a graphics device; it is ignored when the LINEPRINTER option is specified.

The data set specified with the ANNOTATE= option in the PROC MACONTROL statement is a "global" annotate data set in the sense that the information in this data set is displayed on every chart produced in the current run of the MACONTROL procedure.

ANNOTATE2=*SAS-data-set*
ANNO2=*SAS-data-set*

Graphics

specifies an input data set that contains appropriate annotate variables that add features to the trend chart (secondary chart) produced with the TRENDVAR= option in the EWMACHART or MACHART statement.

DATA=*SAS-data-set*

names an input data set that contains raw data (measurements) as observations. If the values of the *subgroup-variable* are numeric, you need to sort the data set so that these values are in increasing order (within BY groups). The DATA= data set can contain more than one observation for each value of the *subgroup-variable*.

You cannot specify a DATA= data set with a HISTORY= or TABLE= data set. If you do not specify an input data set, PROC MACONTROL uses the most recently created data set as a DATA= data set. For more information, see "DATA= Data Set" in the appropriate chart statement chapter.

FORMCHAR(*index***)='***string***'**

Line Printer

defines characters used for features on charts produced on a line printer, where

index

is a list of numbers ranging from 1 to 17. The list identifies which features are controlled with the *string* characters. By default, *index* is omitted, and the FORMCHAR= option gives a *string* for all 17 features.

string

gives characters for features in *index*. Any character or hexadecimal string can be used.

The features associated with values of *index* are as follows:

Value of *index*	Description of Character	Chart Feature
1	vertical bar	frame
2	horizontal bar	frame, central line
3	box character (upper left)	frame
4	box character (upper middle)	serifs, tick (horizontal axis)
5	box character (upper right)	frame
6	box character (middle left)	not used
7	box character (middle middle)	serifs
8	box character (middle right)	tick (vertical axis)
9	box character (lower left)	frame
10	box character (lower middle)	serifs
11	box character (lower right)	frame
12	vertical bar	control limits
13	horizontal bar	control limits
14	box character (upper right)	control limits
15	box character (lower left)	control limits
16	box character (lower right)	control limits
17	box character (upper left)	control limits

Not all printers can produce the characters in the preceding list. By default, the form character list specified by the SAS system FORMCHAR= option is used; otherwise, the default is FORMCHAR='|—-|+|—|====='. If you print to a PC screen or if your device supports the ASCII symbol set (1 or 2), the following is recommended:

```
formchar='B3,C4,DA,C2,BF,C3,C5,B4,C0,C1,D9,BA,CD,BB,C8,BC,D9'X
```

Note that you can use the FORMCHAR= option to temporarily override the values of the SAS system FORMCHAR= option. The values of the SAS system FORMCHAR= option are not altered by the FORMCHAR= option in the PROC MACONTROL statement.

GOUT=*graphics-catalog*

specifies the graphics catalog for graphics output from PROC MACONTROL. This is useful if you want to save the output. The GOUT= option is used only when the chart is created using a graphics device; it is ignored when the LINEPRINTER option is specified. *Graphics*

HISTORY=*SAS-data-set*
HIST=*SAS-data-set*
names an input data set that contains subgroup summary statistics (means, standard deviations, and sample sizes). Typically, this data set is created as an OUTHISTORY= data set in a previous run of PROC MACONTROL or PROC SHEWHART, but it can also be created with a SAS summarization procedure such as PROC MEANS.

If the values of the *subgroup-variable* are numeric, you need to sort the data set so that these values are in increasing order (within BY groups). A HISTORY= data set can contain only one observation for each value for the *subgroup-variable*.

You cannot use a HISTORY= data set with a DATA= or TABLE= data set. If you do not specify an input data set, PROC MACONTROL uses the most recently created data set as a DATA= data set. For more information on HISTORY= data sets, see "HISTORY= Data Set" in the appropriate chart statement chapter.

LIMITS=*SAS-data-set*
names an input data set that contains the control limit parameters for the moving average chart. Each observation in a LIMITS= data set contains the parameters for a *process*.

If you are using Release 6.09 or an earlier release of SAS/QC software, you must specify the options READLIMITS or READINDEX= in the chart statement to read the parameters from the LIMITS= data set. In Release 6.10 and later releases, these options are not needed.

For details about the variables needed in a LIMITS= data set, see "LIMITS= Data Set" in the appropriate chart statement chapter.

If you do not provide a LIMITS= data set, you must specify the parameters with options in the chart statement.

LINEPRINTER
requests that line printer charts be produced. By default, the procedure creates charts for a graphics device.

TABLE=*SAS-data-set*
names an input data set that contains subgroup summary statistics and control limits. Each observation in a TABLE= data set provides information for a particular subgroup and *process*. Typically, this data set is created as an OUTTABLE= data set in a previous run of PROC MACONTROL.

You cannot use a TABLE= data set with a DATA= or HISTORY= data set. If you do not specify an input data set, PROC MACONTROL uses the most recently created data set as a DATA= data set. For more information, see the "TABLE= Data Set" section in the appropriate chart statement chapter.

Input and Output Data Sets

Figure 26.1 summarizes the data sets used with the MACONTROL procedure.

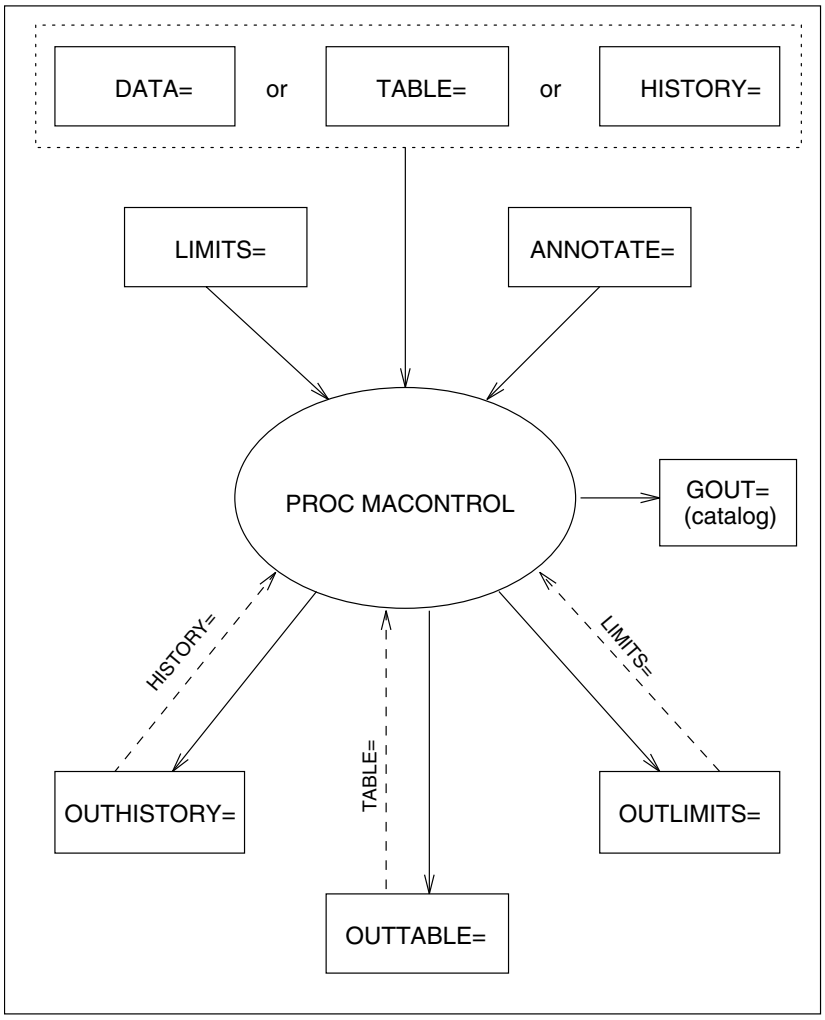


Figure 26.1. Input and Output Data Sets in the MACONTROL Procedure

Chapter 27
EWMACHART Statement

Chapter Contents

OVERVIEW 765

GETTING STARTED 766
- Creating EWMA Charts from Raw Data 766
- Creating EWMA Charts from Subgroup Summary Data 769
- Saving Summary Statistics 771
- Saving Control Limit Parameters 772
- Reading Preestablished Control Limit Parameters 775

SYNTAX 777
- Summary of Options 778
- Dictionary of Special Options 786

DETAILS 790
- Constructing EWMA Charts 790
- Output Data Sets 796
- ODS Tables 798
- Input Data Sets 799
- Methods for Estimating the Standard Deviation 802
- Axis Labels 804
- Missing Values 805

EXAMPLES 805
- Example 27.1. Specifying Standard Values for the Process Mean and Process Standard Deviation 805
- Example 27.2. Displaying Limits Based on Asymptotic Values 807
- Example 27.3. Working with Unequal Subgroup Sample Sizes 808
- Example 27.4. Displaying Individual Measurements on an EWMA Chart . . 813
- Example 27.5. Computing Average Run Lengths 815

Chapter 27
EWMACHART Statement

Overview

The EWMACHART statement creates an exponentially weighted moving average (EWMA) control chart, which is used to determine whether a process is in a state of statistical control and to detect shifts in the process average.

You can use options in the EWMACHART statement to

- specify the weight assigned to the most recent subgroup mean in the computation of the EWMAs
- compute control limits from the data based on a multiple of the standard error of the plotted EWMAs or as probability limits
- tabulate the EWMAs, subgroup sample sizes, subgroup means, subgroup standard deviations, control limits, and other information
- save control limit parameters in an output data set
- save the EWMAs, subgroup sample sizes, subgroup means, and subgroup standard deviations in an output data set
- read control limit parameters from an input data set
- specify one of several methods for estimating the process standard deviation
- specify a known (standard) process mean and standard deviation for computing control limits
- display a secondary chart that plots a time trend removed from the data
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

Getting Started

This section introduces the EWMACHART statement with simple examples that illustrate the most commonly used options. Complete syntax for the EWMACHART statement is presented in the "Syntax" section on page 777, and advanced examples are given in the "Examples" section on page 805.

Creating EWMA Charts from Raw Data

See MACEW1 in the SAS/QC Sample Library

In the manufacture of a metal clip, the gap between the ends of the clip is a critical dimension. To monitor the process for a change in the average gap, subgroup samples of five clips are selected daily. The data are analyzed with an EWMA chart. The gaps recorded during the first twenty days are saved in a SAS data set named CLIPS1.

```
data clips1;
   input day @ ;
   do i=1 to 5;
      input gap @ ;
      output;
      end;
   drop i;
   datalines;
 1  14.76  14.82  14.88  14.83  15.23
 2  14.95  14.91  15.09  14.99  15.13
 3  14.50  15.05  15.09  14.72  14.97
 4  14.91  14.87  15.46  15.01  14.99
 5  14.73  15.36  14.87  14.91  15.25
 6  15.09  15.19  15.07  15.30  14.98
 7  15.34  15.39  14.82  15.32  15.23
 8  14.80  14.94  15.15  14.69  14.93
 9  14.67  15.08  14.88  15.14  14.78
10  15.27  14.61  15.00  14.84  14.94
11  15.34  14.84  15.32  14.81  15.17
12  14.84  15.00  15.13  14.68  14.91
13  15.40  15.03  15.05  15.03  15.18
14  14.50  14.77  15.22  14.70  14.80
15  14.81  15.01  14.65  15.13  15.12
16  14.82  15.01  14.82  14.83  15.00
17  14.89  14.90  14.60  14.40  14.88
18  14.90  15.29  15.14  15.20  14.70
19  14.77  14.60  14.45  14.78  14.91
20  14.80  14.58  14.69  15.02  14.85
;
run;
```

The following statements produce the listing of the data set CLIPS1 shown in Figure 27.1:

```
title 'The Data Set CLIPS1';
proc print data=clips1 noobs;
run;
```

The Data Set CLIPS1

day	gap
1	14.76
1	14.82
1	14.88
1	14.83
1	15.23
2	14.95
2	14.91
2	15.09
2	14.99
2	15.13
.	.
.	.
.	.
20	14.80
20	14.58
20	14.69
20	15.02
20	14.85

Figure 27.1. Partial Listing of the Data Set CLIPS1

The data set CLIPS1 is said to be in "strung-out" form, since each observation contains the day and gap measurement of a single clip. The first five observations contain the gap measurements for the first day, the second five observations contain the gap measurements for the second day, and so on. Because the variable DAY classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable GAP contains the gap measurements and is referred to as the *process variable* (or *process* for short).

The within-subgroup variability of the gap measurements is known to be stable. You can use an EWMA chart to determine whether the mean level is in control. The following statements create the EWMA chart shown in Figure 27.2:

```
symbol h = .8;
title 'EWMA Chart for Gap Measurements';
proc macontrol data=clips1;
   ewmachart gap*day / weight=0.3;
run;
```

This example illustrates the basic form of the EWMACHART statement. After the keyword EWMACHART, you specify the *process* to analyze (in this case, GAP) followed by an asterisk and the *subgroup-variable* (DAY). The WEIGHT= option specifies the weight parameter used to compute the EWMAs. Options such as WEIGHT= are specified after the slash (/) in the EWMACHART statement. A complete list of options is presented in the "Syntax" section on page 777. You must provide the weight parameter to create an EWMA chart. As an alternative to specifying the WEIGHT= option, you can read the weight parameter from an input data set; see "Reading Preestablished Control Limit Parameters" on page 775.

The input data set is specified with the DATA= option in the PROC MACONTROL statement.

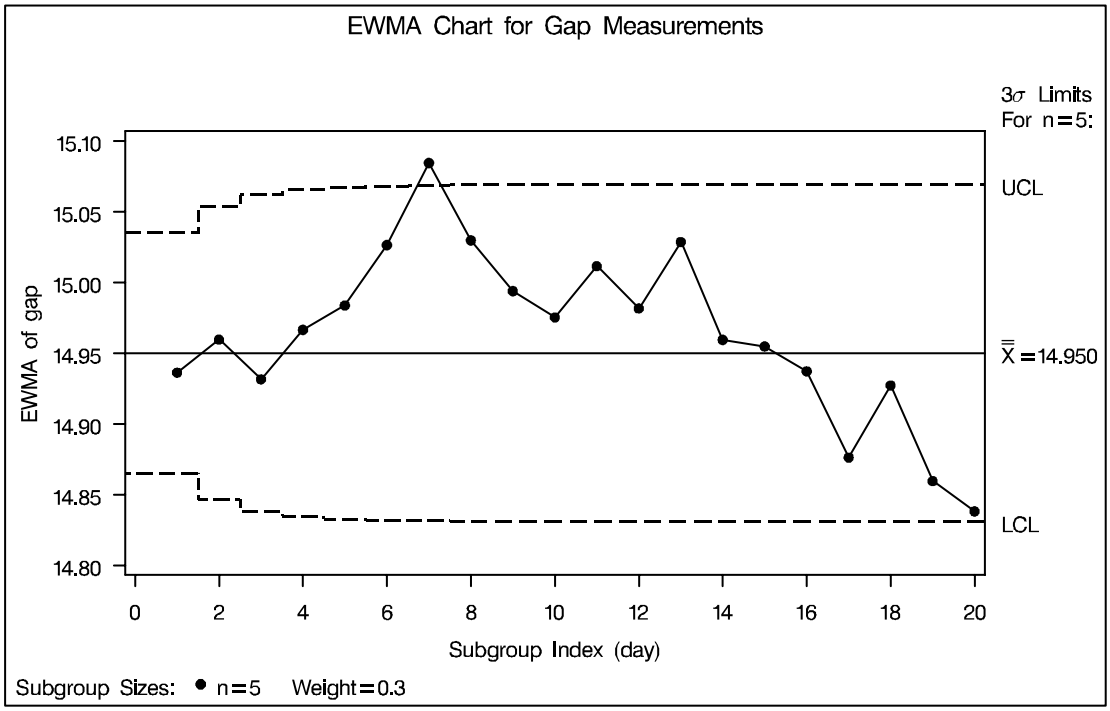


Figure 27.2. Exponentially Weighted Moving Average Chart

Each point on the chart represents the EWMA for a particular day. The EWMA E_1 plotted at DAY=1 is the weighted average of the overall mean and the subgroup mean for DAY=1. The EWMA E_2 plotted at DAY=2 is the weighted average of the EWMA E_1 and the subgroup mean for DAY=2.

$$\begin{aligned} E_1 &= 0.3(14.904) + 0.7(14.952) = 14.9376\text{mm} \\ E_2 &= 0.3(15.014) + 0.7(14.9376) = 14.9605\text{mm} \end{aligned}$$

For succeeding days, the EWMA is the weighted average of the previous EWMA and the present subgroup mean. In the example, a weight parameter of 0.3 is used (since WEIGHT=0.3 is specified in the EWMACHART statement).

Note that the EWMA for the 7th day lies above the upper control limit, signaling an out-of-control process.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 27.19 on page 791.

For computational details, see "Constructing EWMA Charts" on page 790. For more details on reading from a DATA= data set, see "DATA= Data Set" on page 799.

Creating EWMA Charts from Subgroup Summary Data

See MACEW1 in the SAS/QC Sample Library

The previous example illustrates how you can create EWMA charts using raw data (process measurements). However, in many applications the data are provided as subgroup summary statistics. This example illustrates how you can use the EWMACHART statement with data of this type.

The following data set (CLIPSUM) provides the data from the preceding example in summarized form:

```
data clipsum;
   input day gapx gaps;
   gapn=5;
datalines;
 1  14.904  0.18716
 2  15.014  0.09317
 3  14.866  0.25006
 4  15.048  0.23732
 5  15.024  0.26792
 6  15.126  0.12260
 7  15.220  0.23098
 8  14.902  0.17254
 9  14.910  0.19824
10  14.932  0.24035
11  15.096  0.25618
12  14.912  0.16903
13  15.138  0.15928
14  14.798  0.26329
15  14.944  0.20876
16  14.896  0.09965
17  14.734  0.22512
18  15.046  0.24141
19  14.702  0.17880
20  14.788  0.16634
;
run;
```

A partial listing of CLIPSUM is shown in Figure 27.3. There is exactly one observation for each subgroup (note that the subgroups are still indexed by DAY). The variable GAPX contains the subgroup means, the variable GAPS contains the subgroup standard deviations, and the variable GAPN contains the subgroup sample sizes (these are all five).

The Data Set CLIPSUM

day	gapx	gaps	gapn
1	14.904	0.18716	5
2	15.014	0.09317	5
3	14.866	0.25006	5
.	.	.	.
.	.	.	.
.	.	.	.
20	14.788	0.16634	5

Figure 27.3. The Summary Data Set CLIPSUM

You can read this data set by specifying it as a HISTORY= data set in the PROC MACONTROL statement, as follows:

```
title 'EWMA Chart for Gap Measurements';
proc macontrol history=clipsum lineprinter;
   ewmachart gap*day='*' / weight=0.3;
run;
```

The resulting EWMA chart is shown in Figure 27.4. Since the LINEPRINTER * option is specified in the PROC MACONTROL statement, line printer output is produced. The asterisk (*) specified in single quotes after the *subgroup-variable* indicates the character used to plot points. This character must follow an equal sign.

Note that GAP is *not* the name of a SAS variable in the data set but is, instead, the common prefix for the names of the three SAS variables GAPX, GAPS, and GAPN. The suffix characters *X*, *S*, and *N* indicate *mean*, *standard deviation*, and *sample size*, respectively. Thus, you can specify three subgroup summary variables in a HISTORY= data set with a single name (GAP), which is referred to as the *process*. The variables GAPX, GAPS, and GAPN are all required. The name DAY specified after the asterisk is the name of the *subgroup-variable*.

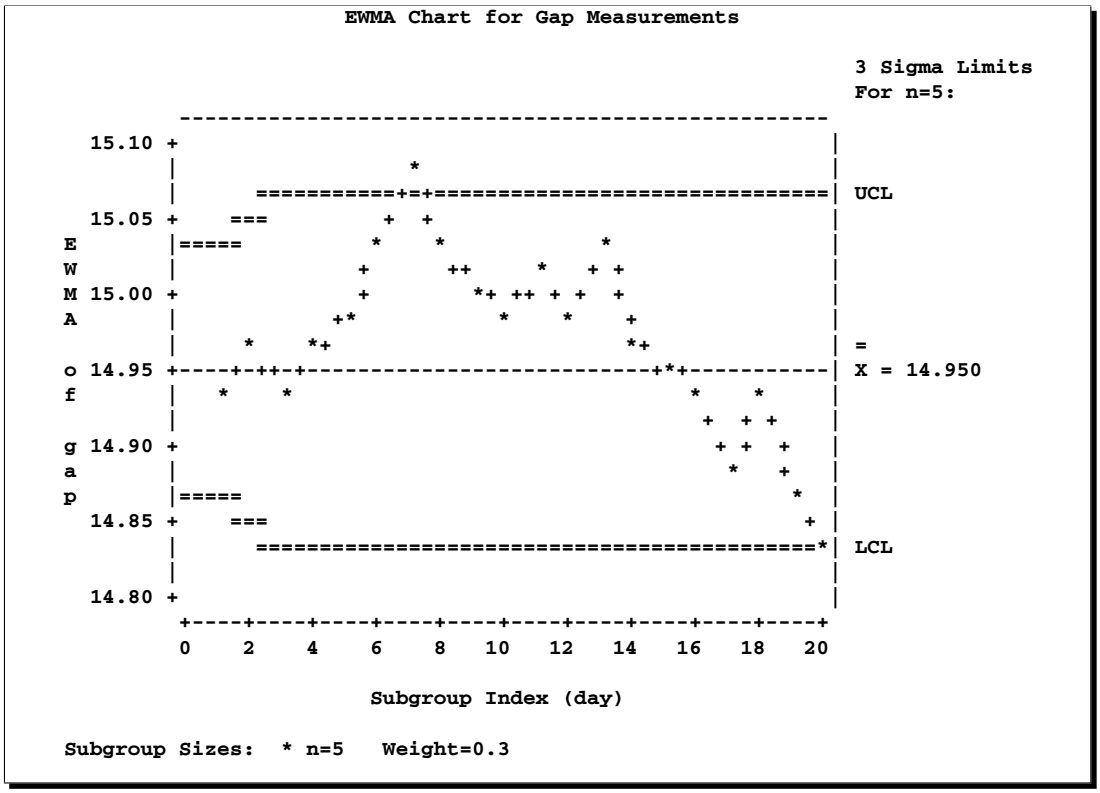


Figure 27.4. EWMA Chart from Summary Data

*In Release 6.12 and previous releases of SAS/QC software, the keyword GRAPHICS was required in the PROC MACONTROL statement to specify that the chart be created with a graphics device. In Version 7, you can specify the LINEPRINTER option to request line printer plots.

In general, a HISTORY= input data set used with the EWMACHART statement must contain the following variables:

- subgroup variable
- subgroup mean variable
- subgroup standard deviation variable
- subgroup sample size variable

Furthermore, the names of subgroup mean, standard deviation, and sample size variables must begin with the *process* name specified in the EWMACHART statement and end with the special suffix characters *X*, *S*, and *N*, respectively. If the names do not follow this convention, you can use the RENAME option in the PROC MACONTROL statement to rename the variables for the duration of the MACONTROL procedure step (see page 1743 for an example of the RENAME option).

In summary, the interpretation of *process* depends on the input data set.

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names of the variables containing the summary statistics.

For more information, see "HISTORY= Data Set" on page 800.

Saving Summary Statistics

In this example, the EWMACHART statement is used to create a summary data set that can be read later by the MACONTROL procedure (as in the preceding example). The following statements read measurements from the data set CLIPS1 and create a summary data set named CLIPHIST:

See MACEW1 in the SAS/QC Sample Library

```
title 'Summary Data Set for Gap Measurements';
proc macontrol data=clips1;
   ewmachart gap*day / weight     = 0.3
                       outhistory = cliphist
                       nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 27.2.

Figure 27.5 contains a partial listing of CLIPHIST.

Summary Data Set for Gap Measurements

day	gapX	gapS	gapE	gapN
1	14.904	0.18716	14.9362	5
2	15.014	0.09317	14.9595	5
3	14.866	0.25006	14.9315	5
4	15.048	0.23732	14.9664	5
5	15.024	0.26792	14.9837	5
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
20	14.788	0.16634	14.8381	5

Figure 27.5. The Summary Data Set CLIPHIST

There are five variables in the data set CLIPHIST.

- DAY contains the subgroup index.
- GAPX contains the subgroup means.
- GAPS contains the subgroup standard deviations.
- GAPE contains the subgroup exponentially weighted moving averages.
- GAPN contains the subgroup sample sizes.

Note that the summary statistic variables are named by adding the suffix characters *X*, *S*, *E*, and *N* to the *process* GAP specified in the EWMACHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 797.

Saving Control Limit Parameters

See MACEW1 in the SAS/QC Sample Library

You can save the control limit parameters for an EWMA chart in a SAS data set; this enables you to use these parameters with future data (see "Reading Preestablished Control Limit Parameters" on page 775) or modify the parameters with a DATA step program.

The following statements read measurements from the data set CLIPS1 (see page 766) and save the control limit parameters in a data set named CLIPLIM:

```
title 'Control Limit Parameters';
proc macontrol data=clips1;
   ewmachart gap*day / weight    = 0.3
                       outlimits = cliplim
                       nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limit parameters, and the NOCHART option suppresses the display of the chart. The data set CLIPLIM is listed in Figure 27.6.

Control Limit Parameters

_VARPER	_SUBGRP_	_TYPE_	_LIMITN_	_MEAN_	_SIGMAS_	_WEIGHT_	_STDDEV_	_WIDTH_
gap	day	ESTIMATE	5	.002699796	3	14.95	0.21108	0.3

Figure 27.6. The Data Set CLIPLIM Containing Control Limit Information

Note that the data set CLIPLIM does not contain the actual control limits but rather the parameters required to compute the limits.

The data set contains one observation with the parameters for *process* GAP. The variable _WEIGHT_ contains the weight parameter used to compute the EWMAs. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates that the values of _MEAN_ and _STDDEV_ are estimates rather than standard values. For more information, see "OUTLIMITS= Data Set" on page 796.

You can create an output data set containing the control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
title 'Summary Statistics and Control Limits';
proc macontrol data=clips1;
   ewmachart gap*day / weight   = 0.3
                       outtable = cliptab
                       nochart;
run;
```

The data set CLIPTAB is listed in Figure 27.7.

Summary Statistics and Control Limits

VAR	day	_SIGMA_	_LIMIT_	_WEIGHT_	_SUBN_	_SUBX_	_SUBS_	_LCLE_	_MEAN_	_UCLE_	_EXLIM_
gap	1	3	5	0.3	5	14.904	0.18716	14.8650	14.9362	14.95	15.0350	
gap	2	3	5	0.3	5	15.014	0.09317	14.8463	14.9595	14.95	15.0537	
gap	3	3	5	0.3	5	14.866	0.25006	14.8383	14.9315	14.95	15.0617	
gap	4	3	5	0.3	5	15.048	0.23732	14.8345	14.9664	14.95	15.0655	
gap	5	3	5	0.3	5	15.024	0.26792	14.8327	14.9837	14.95	15.0673	
gap	6	3	5	0.3	5	15.126	0.12260	14.8319	15.0264	14.95	15.0681	
gap	7	3	5	0.3	5	15.220	0.23098	14.8314	15.0845	14.95	15.0686	UPPER
gap	8	3	5	0.3	5	14.902	0.17254	14.8312	15.0297	14.95	15.0688	
gap	9	3	5	0.3	5	14.910	0.19824	14.8311	14.9938	14.95	15.0689	
gap	10	3	5	0.3	5	14.932	0.24035	14.8311	14.9753	14.95	15.0689	
gap	11	3	5	0.3	5	15.096	0.25618	14.8311	15.0115	14.95	15.0689	
gap	12	3	5	0.3	5	14.912	0.16903	14.8310	14.9816	14.95	15.0690	
gap	13	3	5	0.3	5	15.138	0.15928	14.8310	15.0285	14.95	15.0690	
gap	14	3	5	0.3	5	14.798	0.26329	14.8310	14.9594	14.95	15.0690	
gap	15	3	5	0.3	5	14.944	0.20876	14.8310	14.9548	14.95	15.0690	
gap	16	3	5	0.3	5	14.896	0.09965	14.8310	14.9371	14.95	15.0690	
gap	17	3	5	0.3	5	14.734	0.22512	14.8310	14.8762	14.95	15.0690	
gap	18	3	5	0.3	5	15.046	0.24141	14.8310	14.9271	14.95	15.0690	
gap	19	3	5	0.3	5	14.702	0.17880	14.8310	14.8596	14.95	15.0690	
gap	20	3	5	0.3	5	14.788	0.16634	14.8310	14.8381	14.95	15.0690	

Figure 27.7. The OUTTABLE= Data Set CLIPTAB

This data set contains one observation for each subgroup sample. The variable _EWMA_ contains the EWMAs. The variables _SUBX_, _SUBS_, and _SUBN_ contain the subgroup means, subgroup standard deviations, and subgroup sample sizes, respectively. The variables _LCLE_ and _UCLE_ contain the lower and upper control limits, and the variable _MEAN_ contains the central line. The variables _VAR_ and DAY contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 797.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read CLIPTAB and display a EWMA chart (not shown here) identical to Figure 27.2:

```
title 'EWMA Chart for Gap Measurements';
proc macontrol table=cliptab;
   ewmachart gap*day ;
run;
```

For more information, see "TABLE= Data Set" on page 801.

Reading Preestablished Control Limit Parameters

In the previous example, the OUTLIMITS= data set saved the control limit parameters in the data set CLIPLIM. This example shows how to apply these parameters to new data provided in the following data set:

See MACEW1 in the SAS/QC Sample Library

```
data clips1a;
   label gap='Gap Measurement (mm)';
   input day @;
   do i=1 to 5;
      input gap @;
      output;
   end;
   drop i;
datalines;
21  14.86 15.01 14.67 14.67 15.07
22  14.93 14.53 15.07 15.10 14.98
23  15.27 14.90 15.12 15.10 14.80
24  15.02 15.21 14.93 15.11 15.20
25  14.90 14.81 15.26 14.57 14.94
26  14.78 15.29 15.13 14.62 14.54
27  14.78 15.15 14.61 14.92 15.07
28  14.92 15.31 14.82 14.74 15.26
29  15.11 15.04 14.61 15.09 14.68
30  15.00 15.04 14.36 15.20 14.65
31  14.99 14.76 15.18 15.04 14.82
32  14.90 14.78 15.19 15.06 15.06
33  14.95 15.10 14.86 15.27 15.22
34  15.03 14.71 14.75 14.99 15.02
35  15.38 14.94 14.68 14.77 14.83
36  14.95 15.43 14.87 14.90 15.34
37  15.18 14.94 15.32 14.74 15.29
38  14.91 15.15 15.06 14.78 15.42
39  15.34 15.34 15.41 15.36 14.96
40  15.12 14.75 15.05 14.70 14.74
;
run;
```

The following statements create an EWMA chart for the data in CLIPS1A using the control limit parameters in CLIPLIM:

```
symbol h = .8;
title 'EWMA Chart for Second Set of Gap Measurements';
proc macontrol data=clips1a limits=cliplim;
   ewmachart gap*day;
run;
```

The chart is shown in Figure 27.8.

The LIMITS= option in the PROC MACONTROL statement specifies the data set containing the control limit parameters. By default,* this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name GAP
- the value of _SUBGRP_ matches the *subgroup-variable* name DAY

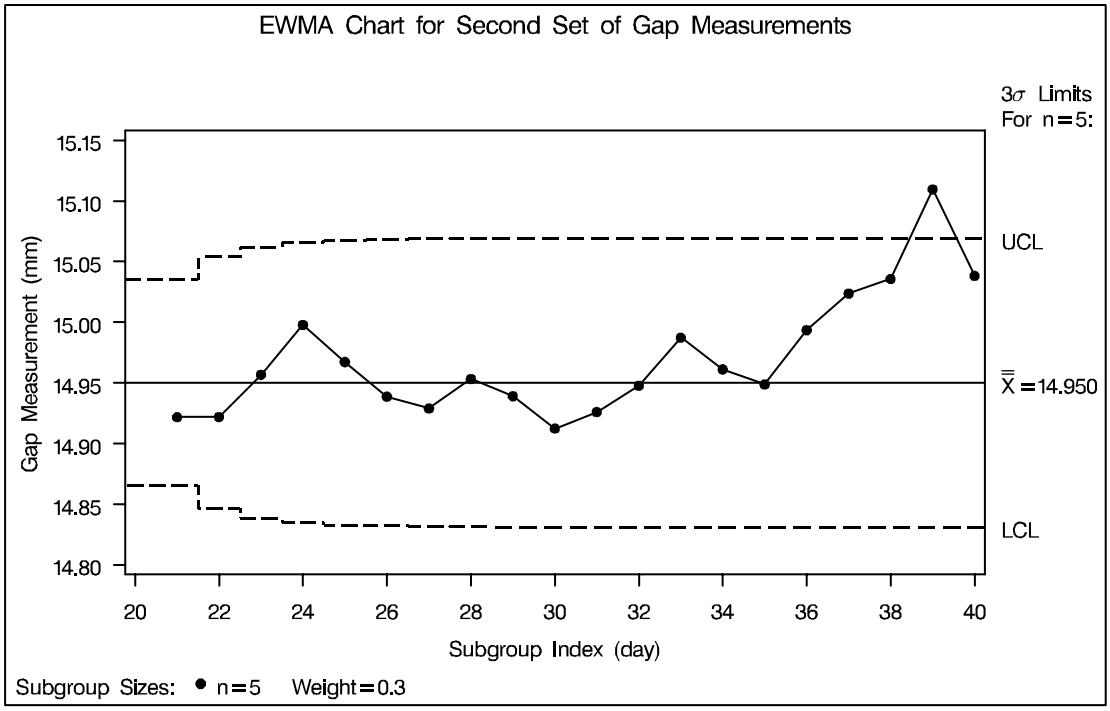


Figure 27.8. EWMA Chart Using Preestablished Control Limit Parameters

Note that the EWMA plotted for the 39^{th} day lies above the upper control limit, signalling an out-of-control process.

In this example, the LIMITS= data set was created in a previous run of the MACONTROL procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 799 for details concerning the variables that you must provide, and see Example 27.1 on page 805 for an illustration.

*In Release 6.09 and in earlier releases, it is also necessary to specify the READLIMITS option to read control limits from a LIMITS= data set.

Syntax

The basic syntax for the EWMACHART statement is as follows:

EWMACHART *process******subgroup-variable* **/ WEIGHT=***value* < *options* > **;**

The general form of this syntax is as follows:

EWMACHART (*processes***)***subgroup-variable* <**(** *block-variables* **)** >
< =*symbol-variable* | ='*character*' > **/ WEIGHT=***value* < *options* >**;**

Note that the WEIGHT= option is required unless its *value* is read from a LIMITS= data set. You can use any number of EWMACHART statements in the MACONTROL procedure. The components of the EWMACHART statement are described as follows.

process
processes
identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC MACONTROL statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating EWMA Charts from Raw Data" on page 766.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating EWMA Charts from Subgroup Summary Data" on page 769.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limit Parameters" on page 772.

A *process* is required. If more than one *process* is specified, enclose the list in parentheses. For example, the following statements request distinct EWMA charts (each using a weight parameter of 0.3) for WEIGHT, LENGTH, and WIDTH:

```
proc macontrol data=measures;
   ewmachart (weight length width)*day / weight=0.3;
run;
```

subgroup-variable
is the variable that classifies the data into subgroups. The *subgroup-variable* is required. In the preceding EWMACHART statement, DAY is the subgroup variable. For details, see "Subgroup Variables" on page 1771.

block-variables
are optional variables that group the data into blocks of consecutive subgroups. The

blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 1932 for an example.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker or plotting character used to plot the EWMAs.

- If you produce a chart on a line printer, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce a chart on a graphics device, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOLn statements. See "Displaying Stratification in Levels of a Classification Variable" on page 1931 for an example.

character
specifies a plotting character for charts produced on line printers. For example, the following statements create an EWMA chart using an asterisk (*) to plot the points:

```
proc macontrol data=values;
   ewmachart length*hour='*' / weight=0.3;
run;
```

options
specify chart parameters, enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The "Summary of Options" section, which follows, lists all options by function.

Summary of Options

The following tables list the EWMACHART statement options by function. Options unique to the MACONTROL procedure are listed in Table 27.1 and Table 27.2, and they are described in detail in "Dictionary of Special Options" on page 786. Options that are common to both the MACONTROL and SHEWHART procedures are listed in Table 27.3 to Table 27.18. They are described in detail beginning on page 1851.

Table 27.1. Options for Specifying Exponentially Weighted Moving Average Charts

ALPHA=*value*	requests probability limits for control charts
ASYMPTOTIC	requests constant control limits based on asymptotic expressions
LIMITN=n|VARYING	specifies either a fixed nominal sample size (n) for control limits or allows the control limits to vary with subgroup sample size
MU0=*value*	specifies a standard (known) value μ_0 for the process mean
NOREADLIMITS	specifies that control limit parameters are not to be read from a LIMITS= data set (Release 6.10 and later releases)
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from the LIMITS= data set when both variables are available
READINDEX='*value*'	reads control limit parameters from the first observation in the LIMITS= data set where the variable _INDEX_ equals *value*
READLIMITS	reads control limit parameters from a LIMITS= data set (Release 6.09 and earlier releases)
RESET	requests that the value of the EWMA be reset after each out-of-control point
SIGMA0=*value*	specifies standard (known) value σ_0 for process standard deviation
SIGMAS=k	specifies width of control limits in terms of multiple k of standard error of plotted EWMAs
WEIGHT=*value*	specifies weight assigned to the most recent subgroup mean in the computation of the EWMA

Table 27.2. Options for Plotting Subgroup Means

CMEANSYMBOL=*color*	specifies color for MEANSYMBOL= symbol
MEANCHAR='*character*'	specifies *character* to plot subgroup means on line printer
MEANSYMBOL= *keyword*	specifies symbol to plot subgroup means on graphics device

Table 27.3. Tabulation Options

TABLE	creates a basic table of subgroup variable values, subgroup sample sizes, subgroup means, subgroup EWMAs, and control limits
TABLEALL	equivalent to the options TABLE, TABLECENTRAL, TABLEID, and TABLEOUT
TABLECENTRAL	augments basic table with the value of the central line
TABLEID	augments basic table with columns for ID variables
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded

Note that specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points.

Table 27.4. Axis and Axis Label Options

CAXIS=*color*	specifies color for axis lines and tick marks
CFRAME=*color*|(*color-list*)	specifies fill colors for frame for plot area
CTEXT=*color*	specifies color for tick mark values and axis labels
HAXIS=*values*|AXIS*n*	specifies major tick mark values for horizontal axis
HEIGHT=*value*	specifies height of axis label and axis legend text
HMINOR=*n*	specifies minor tick marks between major horizontal tick marks
HOFFSET=*value*	specifies length of offset at both ends of horizontal axis
INTSTART=*value*	specifies first major tick mark value for numeric horizontal axis
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=*n*	specifies thinning factor for tick mark labels on horizontal axis
SPLIT='*character*'	specifies splitting character for axis labels
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=*values*|AXIS*n*	specifies major tick mark values for vertical axis on EWMA chart
VAXIS2=*values*|AXIS*n*	specifies major tick mark values for vertical axis on trend chart
VFORMAT=*format*	specifies format for tick mark labels on vertical axis of EWMA chart
VFORMAT2=*format*	specifies format for tick mark labels on vertical axis of trend chart
VMINOR=*n*	specifies minor tick marks between major vertical tick marks
VOFFSET=*value*	specifies length of offset at both ends of vertical axis
WAXIS=*n*	specifies width of axis lines

Table 27.5. Process Mean and Standard Deviation Options

SMETHOD=*keyword*	specifies method for estimating process standard deviation σ
TYPE=*keyword*	identifies whether parameters are estimates or standard values and specifies value of _TYPE_ in OUTLIMITS= data set

Table 27.6. Grid Options

CGRID=*color*	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to chart
LENDGRID=*linetype*	specifies line type for grid requested with the ENDGRID option
LGRID=*linetype*	specifies line type for grid requested with the GRID option
WGRID=*n*	specifies width of grid lines

Table 27.7. Reference Line Options

CHREF=*color*	specifies color for HREF= and HREF2= lines
CVREF=*color*	specifies color for VREF= and VREF2= lines
HREF=*values*|*SAS-data-set*	specifies reference lines perpendicular to horizontal axis on EWMA chart
HREF2=*values*|*SAS-data-set*	specifies reference lines perpendicular to horizontal axis on trend chart
HREFCHAR='*character*'	specifies line character for HREF= and HREF2= lines
HREFDATA=*SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on EWMA chart
HREF2DATA=*SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFLABELS='*label1*'...'*labeln*'	specifies labels for HREF= lines
HREF2LABELS='*label1*'...'*labeln*'	specifies labels for HREF2= lines
HREFLABPOS=*n*	specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=*linetype*	specifies line type for HREF= and HREF2= lines
LVREF=*linetype*	specifies line type for VREF= and VREF2= lines
NOBYREF	specifies that reference line information in a data set is to be applied uniformly to charts created for all BY groups
VREF=*values*|*SAS-data-set*	specifies reference lines perpendicular to vertical axis on EWMA chart
VREF2=*values*|*SAS-data-set*	specifies reference lines perpendicular to vertical axis on trend chart
VREFCHAR='*character*'	specifies line character for VREF= and VREF2= lines
VREFLABELS='*label1*'...'*labeln*'	specifies labels for VREF= lines
VREF2LABELS='*label1*'...'*labeln*'	specifies labels for VREF2= lines
VREFLABPOS=*n*	specifies position of VREFLABELS= and VREF2LABELS= labels

Table 27.8. Block Variable Legend Options

BLOCKLABELPOS=*keyword*	specifies position of label for *block-variable* legend
BLOCKLABTYPE=*value*|*keyword*	specifies text size of *block-variable* legend
BLOCKPOS=*n*	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=*color-list*	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=*variable*|(*variables*)	specifies one or more variables whose values are colors for filling background of *block-variable* legend

Table 27.9. Options for Displaying Control Limits

CINFILL=*color*	specifies color for area inside control limits
CLIMITS=*color*	specifies color of control limits, central line, and related labels
LCLLABEL='*label*'	specifies label for lower control limit
LIMLABSUBCHAR='*character*'	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=*linetype*	line type for control limits
NDECIMAL=*n*	specifies number of digits to right of decimal place in default labels for control limits and central line
NOCTL	suppresses display of central line
NOLCL	suppresses display of lower control limit
NOLIMITLABEL	suppresses labels for control limits and center line
NOLIMITS	suppresses display of control limits
NOLIMITSLEGEND	suppresses legend for control limits
NOUCL	suppresses display of upper control limit
UCLLABEL='*string*'	specifies label for upper control limit
WLIMITS=*n*	width for control limits and central line
XSYMBOL='*string*'| *keyword*	specifies label for central line

Table 27.10. Options for Interactive Control Charts

HTML=(*variable*)	specifies a variable whose values are URLs to be associated with subgroups
HTML2=(*variable*)	specifies variable whose values are URLs to be associated with points on trend chart
HTML_LEGEND=(*variable*)	specifies a variable whose values are URLs to be associated with symbols in the symbol legend
WEBOUT=*SAS-data-set*	creates an OUTTABLE= data set with additional graphics co-ordinate data

Table 27.11. Options for Plotting and Labeling Points

ALLLABEL=VALUE|(*variable*)	labels every point on EWMA chart
ALLLABEL2=VALUE|(*variable*)	labels every point on trend chart
CCONNECT=*color*	specifies color for line segments that connect points on chart
CFRAMELAB=*color*	specifies fill color for frame around labeled points
CLABEL=*color*	specifies color for labels
CNEEDLES=*color*	specifies color for needles that connect points to central line
CONNECTCHAR='*character*'	specifies character used to form line segments that connect points on EWMA chart
COUT=*color*	specifies color for line segments that connect points exceeding control limits
COUTFILL=*color*	specifies color for areas between connected points and control limits
LABELANGLE=*angle*	specifies angle at which labels are drawn
LABELFONT=*font*	specifies a software font for labels requested by the ALLLABEL=, ALLLABEL2=, OUTLABEL=, and STARLABEL= options
LABELHEIGHT=*font*	specifies the height (in vertical percent screen units) for labels requested by the ALLLABEL=, ALLLABEL2=, OUTLABEL=, and STARLABEL= options
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on EWMA chart
NOTRENDCONNECT	suppresses line segments that connect points on trend chart
OUTLABEL=VALUE|(*variable*)	labels points exceeding control limits
SYMBOLCHARS='*characters*'	specifies characters indicating *symbol-variable*
SYMBOLLEGEND=NONE|*name*	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=*keyword*	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL	turns point labels so that they are strung out vertically

Table 27.12. Input Data Set Options

MISSBREAK	specifies that observations with missing values are not to be processed

Table 27.13. Output Data Set Options

OUTHISTORY= *SAS-data-set*	creates output data set containing subgroup summary statistics
OUTINDEX='*string*'	specifies value of the variable _INDEX_ in OUTLIMITS= data set
OUTLIMITS= *SAS-data-set*	creates output data set containing control limit parameters
OUTPHASE='*string*'	specifies value of the variable _PHASE_ in OUTHISTORY= or OUTTABLE= data set
OUTTABLE= *SAS-data-set*	creates output data set containing subgroup summary statistics and control limits

Table 27.14. Plot Layout Options

ALLN	plots EWMA for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts only when exceptions occur
INTERVAL=*keyword*	specifies natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=*n*	specifies maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of EWMA chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=*n*	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=*n*	specifies number of pages or screens to be used to display chart
TRENDVAR=*variable*| (*variable-list*)	specifies list of trend variables
YPCT1=*value*	specifies length of vertical axis on EWMA chart as a percentage of sum of lengths of vertical axes for EWMA and trend charts
ZEROSTD	displays $\bar{X}$ chart regardless of whether $\hat{\sigma} = 0$

Table 27.15. Phase Options

CPHASELEG=*color*	specifies text color for *phase* legend
OUTPHASE='*string*'	specifies value of _PHASE_ in OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=*value*| *keyword*	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASEREF	delineates *phases* with vertical reference lines
READPHASES= ALL| '*label1*'...'*labeln*'	specifies *phases* to be read from input data set

Table 27.16. Graphical Enhancement Options

ANNOTATE= *SAS-data-set*	specifies annotate data set that adds features to EWMA chart
ANNOTATE2= *SAS-data-set*	specifies annotate data set that adds features to trend chart
DESCRIPTION='*string*'	specifies string that appears in the description field of PROC GREPLAY master menu for EWMA chart
FONT=*font*	specifies software font for labels and legends on chart
NAME='*string*'	specifies name that appears in the name field of the PROC GREPLAY master menu for EWMA chart
PAGENUM='*string*'	specifies the form of the label used in pagination
PAGENUMPOS= *keyword*	specifies the position of the page number requested with the PAGENUM= option
WTREND=*n*	specifies width of line segments connecting points on trend chart

Table 27.17. Clipping Options

CCLIP=*color*	color for plot symbol for clipped points
CLIPCHAR='*character*'	plot character for clipped points
CLIPFACTOR=*value*	determines extent to which extreme points are clipped
CLIPLEGEND='*string*'	text for clipping legend
CLIPLEGPOS=*keyword*	position of clipping legend
CLIPSUBCHAR= '*character*'	substitution character for CLIPLEGEND= text
CLIPSYMBOL=*symbol*	plot symbol for clipped points

Table 27.18. Star Options

CSTARCIRCLES=*color*	specifies color for STARCIRCLES= circles
CSTARFILL=*color*| *(variable)*	specifies color for filling stars
CSTAROUT=*color*	specifies outline color for stars exceeding inner or outer circles
CSTARS=*color*| *(variable)*	specifies color for outlines of stars
LSTARCIRCLES= *linetypes*	specifies line types for STARCIRCLES= circles
LSTARS=*linetype*| *(variable)*	specifies line types for outlines of stars requested with the STARVERTICES= option
STARBDRADIUS=*value*	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=*value-list*	specifies reference circles for stars
STARINRADIUS=*value*	specifies inner radius of stars
STARLABEL=*keyword*	specifies vertices to be labeled
STARLEGEND=*keyword*	specifies style of legend for star vertices
STARLEGENDLAB= *'label'*	specifies label for STARLEGEND= legend
STAROUTRADIUS=*value*	specifies outer radius of stars
STARSPEC=*value*| *SAS-data-set*	specifies method used to standardize vertex variables
STARSTART=*value*	specifies angle for first vertex
STARTYPE=*keyword*	specifies graphical style of star
STARVERTICES= *variable*|*(variables)*	superimposes star at each point on EWMA chart
WSTARCIRCLES=*n*	specifies width of STARCIRCLES= circles
WSTARS=*n*	specifies width of STARVERTICES= stars

Dictionary of Special Options

The marginal notes *Graphics* and *Line Printer* identify options that apply to graphics devices and line printers, respectively.

ALPHA=*value*

requests *probability limits*. If you specify ALPHA=α, the control limits are computed so that the probability is α that a single EWMA exceeds its control limits. The value of α can range between 0 and 1. This assumes that the process is in statistical control and that the data follow a normal distribution. For the equations used to compute probability limits, see "Control Limits" on page 791.

Note the following:

- As an alternative to specifying ALPHA=α, you can read α from the variable _ALPHA_ in a LIMITS= data set by specifying the READALPHA option.
- As an alternative to specifying ALPHA=α (or reading _ALPHA_ from a LIMITS= data set), you can request "$k\sigma$ control limits" by specifying SIGMAS=k (or reading _SIGMAS_ from a LIMITS= data set).

If you specify neither the ALPHA= option nor the SIGMAS= option, the procedure computes 3σ control limits by default.

ASYMPTOTIC
requests constant upper and lower control limits based on the following asymptotic expressions:

$$\text{LCL} = \overline{\overline{X}} - k\hat{\sigma}\sqrt{r/n(2-r)}$$

$$\text{UCL} = \overline{\overline{X}} + k\hat{\sigma}\sqrt{r/n(2-r)}$$

Here r is the weight parameter $(0 < r \leq 1)$, and n is the nominal sample size associated with the control limits. Substitute $\Phi^{-1}(1-\alpha/2)$ for k if you specify probability limits with the ALPHA= option. When you do not specify the ASYMPTOTIC option, the control limits are computed using the exact formulas in Table 27.19 on page 791. Use the ASYMPTOTIC option only if all the subgroup sample sizes are the same or if you specify LIMITN=n. See Example 27.2 on page 807.

CMEANSYMBOL=*color*
specifies the *color* for the symbol requested with the MEANSYMBOL= option. The default *color* is the first color in the device color list. *Graphics*

LIMITN=*n*
LIMITN=VARYING
specifies either a fixed or varying nominal sample size for the control limits.

If you specify LIMITN=n, EWMAs are calculated and displayed only for those subgroups with a sample size equal to n, unless you also specify the ALLN option, which causes all the EWMAs to be calculated and displayed. By default (or if you specify LIMITN=VARYING), EWMAs are calculated and displayed for all subgroups, regardless of sample size.

MEANCHAR='*character***'**
specifies a *character* used to plot the subgroup mean for each subgroup. By default, subgroup means are not plotted. *Line Printer*

MEANSYMBOL=*keyword*
specifies a symbol used to plot the subgroup mean for each subgroup. By default, subgroup means are not plotted. *Graphics*

MU0=*value*
specifies a known (standard) value μ_0 for the process mean μ. By default, μ is estimated from the data. See Example 27.1 on page 805.

Note: As an alternative to specifying MU0=μ_0, you can read a predetermined value for μ_0 from the variable _MEAN_ in a LIMITS= data set.

NOREADLIMITS
specifies that control limit parameters for each *process* listed in the EWMACHART statement are *not* to be read from the LIMITS= data set specified in the PROC

MACONTROL statement. The NOREADLIMITS option is available only in Release 6.10 and later releases.

The following example illustrates the NOREADLIMITS option:

```
proc macontrol data=pistons limits=diamlim;
   ewmachart diameter*hour;
   ewmachart diameter*hour / noreadlimits weight=0.3;
run;
```

The first EWMACHART statement reads the control limits from the first observation in the data set DIAMLIM for which the variable _VAR_ is equal to `diameter` and the variable _SUBGRP_ is equal to `hour`. The second EWMACHART statement computes estimates of the process mean and standard deviation for the control limits from the measurements in the data set PISTONS. Note that the second EWMACHART statement is equivalent to the following statements, which would be more commonly used:

```
proc macontrol data=pistons;
   ewmachart diameter*hour / weight=0.3;
run;
```

For more information about reading control limit parameters from a LIMITS= data set, see the READLIMITS option later in this list.

READALPHA
specifies that the variable _ALPHA_, rather than the variable _SIGMAS_, is to be read from a LIMITS= data set when both variables are available in the data set. Thus the limits displayed are probability limits. If you do not specify the READALPHA option, then _SIGMAS_ is read by default.

READINDEX=*'value'*
reads control limit parameters from a LIMITS= data set (specified in the PROC MACONTROL statement) for each *process* listed in the EWMACHART statement.

The control limit parameters for a particular *process* are read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches *process*
- the value of _SUBGRP_ matches the *subgroup-variable*
- the value of _INDEX_ matches *value*

The *value* can be up to 48 characters and must be enclosed in quotes.

READLIMITS
specifies that control limit parameters are to be read from a LIMITS= data set specified in the PROC MACONTROL statement. The parameters for a particular *process* are read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches *process*
- the value of _SUBGRP_ matches the *subgroup variable*

The use of the READLIMITS option depends on which release of SAS/QC software you are using.

- **In Release 6.10 and later releases, the READLIMITS option is not necessary.** To read control limits parameters as described previously, you simply specify a LIMITS= data set. However, even though the READLIMITS option is redundant, it continues to function as in earlier releases.
- **In Release 6.09 and earlier releases, you must specify the READLIMITS option to read control limits parameters as described previously.** If you specify a LIMITS= data set without specifying the READLIMITS option (or the READINDEX= option), the control limits are computed from the data and the value of the weight parameter is specified with the WEIGHT= option.

RESET
requests that the value of the EWMA be reset after each out-of-control point. Specifically, when a point exceeds the control limits, the EWMA for the next subgroup is computed as the weighted average of the subgroup mean and the overall mean. By default, the EWMAs are not reset.

SIGMA0=*value*
specifies a known (standard) value σ_0 for the process standard deviation σ. The *value* must be positive. By default, the MACONTROL procedure estimates σ from the data using the formulas given in "Methods for Estimating the Standard Deviation" on page 802.

Note: As an alternative to specifying SIGMA0=σ_0, you can read a predetermined value for σ_0 from the variable _STDDEV_ in a LIMITS= data set.

SIGMAS=*value*
specifies the width of the control limits in terms of the multiple k of the standard error of the plotted EWMAs on the chart. The value of k must be positive. By default, $k = 3$ and the control limits are 3σ limits.

WEIGHT=*value*
specifies the weight r assigned to the most recent subgroup mean in the computation of the EWMA ($0 < r \leq 1$). The WEIGHT= option is required unless you read control limit parameters from a LIMITS= data set or a TABLE= data set. See "Choosing the Value of the Weight Parameter" on page 792 for details.

Details

Constructing EWMA Charts

The following notation is used in this section:

E_i	exponentially weighted moving average for the i^{th} subgroup
r	EWMA weight parameter $(0 < r \leq 1)$
μ	process mean (expected value of the population of measurements)
σ	process standard deviation (standard deviation of the population of measurements)
x_{ij}	j^{th} measurement in i^{th} subgroup, with j =1, 2, 3, . . . , n_i
n_i	sample size of i^{th} subgroup
$\overline{X}_i$	mean of measurements in i^{th} subgroup. If $n_i = 1$, then the subgroup mean reduces to the single observation in the subgroup
$\overline{\overline{X}}$	weighted average of subgroup means
$\Phi^{-1}(\cdot)$	inverse standard normal function

Plotted Points

Each point on the chart indicates the value of the exponentially weighted moving average (EWMA) for that subgroup. The EWMA for the i^{th} subgroup (E_i) is defined recursively as

$$E_i = r\overline{X}_i + (1 - r)E_{i-1}\,, \quad i > 0$$

where r is a weight parameter $(0 < r \leq 1)$. Some authors (for example, Hunter 1986 and Crowder 1987a,b) use the symbol λ instead of r for the weight. You can specify the weight with the WEIGHT= option in the EWMACHART statement or with the variable _WEIGHT_ in a LIMITS= data set. If you specify a known value (μ_0) for μ, $E_0 = \mu_0$; otherwise, $E_0 = \overline{\overline{X}}$.

The preceding equation can be rewritten as

$$E_i = E_{i-1} + r(\overline{X}_i - E_{i-1})$$

which expresses the current EWMA as the previous EWMA plus the weighted error in the prediction of the current mean based on the previous EWMA.

The EWMA for the i^{th} subgroup can also be written as

$$E_i = r\sum_{j=0}^{i-1}(1 - r)^j\overline{X}_{i-j} + (1 - r)^iE_0$$

which expresses the EWMA as a weighted average of past subgroup means, where the weights decline exponentially, and the heaviest weight is assigned to the most recent subgroup mean.

Central Line

By default, the central line on an EWMA chart indicates an estimate for μ, which is computed as

$$\hat{\mu} = \overline{\overline{X}} = \frac{n_1\bar{X}_1 + \cdots + n_N\bar{X}_N}{n_1 + \cdots + n_N}$$

If you specify a known value (μ_0) for μ, the central line indicates the value of μ_0.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of E_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that E_i exceeds the limits

The following table presents the formulas for the limits:

Table 27.19. Limits for an EWMA Chart

Control Limits
LCL = lower limit = $\overline{\overline{X}} - k\hat{\sigma}r\sqrt{\sum_{j=0}^{i-1}(1-r)^{2j}/n_{i-j}}$
UCL = upper limit = $\overline{\overline{X}} + k\hat{\sigma}r\sqrt{\sum_{j=0}^{i-1}(1-r)^{2j}/n_{i-j}}$

Probability Limits
LCL = lower limit = $\overline{\overline{X}} - \Phi^{-1}(1-\alpha/2)\hat{\sigma}r\sqrt{\sum_{j=0}^{i-1}(1-r)^{2j}/n_{i-j}}$
UCL = upper limit = $\overline{\overline{X}} + \Phi^{-1}(1-\alpha/2)\hat{\sigma}r\sqrt{\sum_{j=0}^{i-1}(1-r)^{2j}/n_{i-j}}$

These formulas assume that the data are normally distributed. If standard values μ_0 and σ_0 are available for μ and σ, respectively, replace $\overline{\overline{X}}$ with μ_0 and $\hat{\sigma}$ with σ_0 in Table 27.19. Note that the limits vary with both n_i and i.

If the subgroup sample sizes are constant ($n_i = n$), the formulas for the control limits simplify to

$$\text{LCL} = \overline{\overline{X}} - k\hat{\sigma}\sqrt{r(1-(1-r)^{2i})/n(2-r)}$$

$$\text{UCL} = \overline{\overline{X}} + k\hat{\sigma}\sqrt{r(1-(1-r)^{2i})/n(2-r)}$$

Consequently, when the subgroup sample sizes are constant, the width of the control limits increases monotonically with i. For probability limits, replace k with $\Phi^{-1}(1-\alpha/2)$ in the previous equations. Refer to Roberts (1959) and Montgomery (1996).

As i becomes large, the upper and lower control limits approach constant values:

$$\text{LCL} = \overline{\overline{X}} - k\hat{\sigma}\sqrt{r/n(2-r)}$$

$$\text{UCL} = \overline{\overline{X}} + k\hat{\sigma}\sqrt{r/n(2-r)}$$

Some authors base the control limits for EWMA charts on the asymptotic expressions in the two previous equations. For asymptotic probability limits, replace k with $\Phi^{-1}(1-\alpha/2)$ in these equations. You can display asymptotic limits by specifying the ASYMPTOTIC option.

Uniformly weighted moving average charts and exponentially weighted moving average charts have similar properties, and their asymptotic control limits are identical provided that

$$r = 2/(w+1)$$

where w is the weight factor for uniformly weighted moving average charts. Refer to Wadsworth and others (1986) and the *ASQC Glossary and Tables for Statistical Quality Control* (1983).

You can specify parameters for the EWMA limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify r with the WEIGHT= option or with the variable _WEIGHT_ in a LIMITS= data set.
- Specify μ_0 with the MU0= option or with the variable _MEAN_ in a LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in a LIMITS= data set.

Choosing the Value of the Weight Parameter

Various approaches have been proposed for choosing the value of r.

- Hunter (1986) states that the choice "can be left to the judgment of the quality control analyst" and points out that the smaller the value of r, "the greater the influence of the historical data."

- Hunter (1986) also discusses a least squares procedure for estimating r from the data, **assuming an exponentially weighted moving average model for the data**. In this context, the fitted EWMA model provides a forecast of the process that is the basis for dynamic process control. You can use the ARIMA procedure in SAS/ETS software to compute the least squares estimate of r. (Refer to *SAS/ETS User's Guide* for information on PROC ARIMA.) Also see "Autocorrelation in Process Data" on page 2001.
- A number of authors have studied the design of EWMA control schemes based on average run length (ARL) computations. The ARL is the expected number of points plotted before a shift is detected. Ideally, the ARL should be short when a shift occurs, and it should be long when there is no shift (the process is in control.) The effect of r on the ARL was described by Roberts (1959), who used simulation methods. The ARL function was approximated and tabulated by Robinson and Ho (1978), and a more general method for studying run-length distributions of EWMA charts was given by Crowder (1987a,b). Unlike Hunter (1986), these authors assume the data are independent and identically distributed; typically the normal distribution is assumed for the data, although the methods extend to nonnormal distributions. A more detailed discussion of the ARL approach follows.

Average run lengths for two-sided EWMA charts are shown in Table 27.20, which is patterned after Table 1 of Crowder (1987a,b). The ARLs were computed using the EWMAARL DATA step function (see page 2103 for details on the EWMAARL function). Note that Crowder (1987a,b) uses the notation L in place of k and the notation λ in place of r.

You can use Table 27.20 to find a combination of k and r that yields a desired ARL for an in-control process ($\delta = 0$) and for a specified shift of δ. Note that δ is assumed to be standardized; in other words, if a shift of Δ is to be detected in the process mean μ, and if σ is the process standard deviation, you should select the table entry with

$$\delta = \Delta/(\sigma/\sqrt{n})$$

where n is the subgroup sample size. Thus, δ can be regarded as the shift in the sampling distribution of the subgroup mean.

For example, suppose you want to construct an EWMA scheme with an in-control ARL of 90 and an ARL of 9 for detecting a shift of $\delta = 1$. Table 27.20 shows that the combination $r = 0.5$ and $k = 2.5$ yields an in-control ARL of 91.17 and an ARL of 8.27 for $\delta = 1$.

Crowder (1987a,b) cautions that setting the in-control ARL at a desired level does not guarantee that the probability of an early false signal is acceptable. For further details concerning the distribution of the ARL, refer to Crowder (1987a,b).

In addition to using Table 27.20 or the EWMAARL DATA step function to choose a EWMA scheme with desired average run length properties, you can use them to evaluate an existing EWMA scheme. For example, the "Getting Started" section of this chapter contains EWMA schemes with $r = 0.3$ and $k = 3$. The following statements

use the EWMAARL function to compute the in-control ARL and the ARLs for shifts of $\delta = 0.25$ and $\delta = 0.5$:

```
data arlewma;
   arlin = ewmaarl(  0,0.3,3.0);
   arl1  = ewmaarl(.25,0.3,3.0);
   arl2  = ewmaarl(.50,0.3,3.0);
run;
```

The in-control ARL is 465.553, the ARL for $\delta = .25$ is 178.741, and the ARL for $\delta = .5$ is 53.1603. See Example 27.5 on page 815 for an illustration of how to use the EWMAARL function to compute average run lengths for various EWMA schemes and shifts.

Table 27.20. Average Run Lengths for Two-Sided EWMA Charts

		r (weight parameter)					
k	δ	0.05	0.10	0.25	0.50	0.75	1.00
2.0	0.00	127.53	73.28	38.56	26.45	22.88	21.98
2.0	0.25	43.94	34.49	24.83	20.12	18.86	19.13
2.0	0.50	18.97	15.53	12.74	11.89	12.34	13.70
2.0	0.75	11.64	9.36	7.62	7.29	7.86	9.21
2.0	1.00	8.38	6.62	5.24	4.91	5.26	6.25
2.0	1.25	6.56	5.13	3.96	3.59	3.76	4.40
2.0	1.50	5.41	4.20	3.19	2.80	2.84	3.24
2.0	1.75	4.62	3.57	2.68	2.29	2.26	2.49
2.0	2.00	4.04	3.12	2.32	1.95	1.88	2.00
2.0	2.25	3.61	2.78	2.06	1.70	1.61	1.67
2.0	2.50	3.26	2.52	1.85	1.51	1.42	1.45
2.0	2.75	2.99	2.32	1.69	1.37	1.29	1.29
2.0	3.00	2.76	2.16	1.55	1.26	1.19	1.19
2.0	3.25	2.56	2.03	1.43	1.18	1.13	1.12
2.0	3.50	2.39	1.93	1.32	1.12	1.08	1.07
2.0	3.75	2.26	1.83	1.24	1.08	1.05	1.04
2.0	4.00	2.15	1.73	1.17	1.05	1.03	1.02
2.5	0.00	379.09	223.35	124.18	91.17	82.49	80.52
2.5	0.25	73.98	66.59	59.66	58.33	61.07	65.77
2.5	0.50	26.63	23.63	23.28	27.16	33.26	41.49
2.5	0.75	15.41	12.95	11.96	13.96	18.05	24.61
2.5	1.00	10.79	8.75	7.52	8.27	10.57	14.92
2.5	1.25	8.31	6.60	5.39	5.52	6.75	9.46
2.5	1.50	6.78	5.31	4.18	4.03	4.65	6.30
2.5	1.75	5.75	4.46	3.43	3.14	3.43	4.41
2.5	2.00	5.00	3.86	2.92	2.57	2.67	3.24
2.5	2.25	4.43	3.42	2.56	2.18	2.17	2.49
2.5	2.50	4.00	3.07	2.29	1.90	1.83	2.00
2.5	2.75	3.64	2.80	2.08	1.69	1.59	1.67

Table 27.20. (continued)

k	δ	0.05	0.10	0.25	0.50	0.75	1.00
2.5	3.00	3.36	2.57	1.91	1.52	1.41	1.45
2.5	3.25	3.12	2.39	1.77	1.39	1.29	1.29
2.5	3.50	2.92	2.24	1.64	1.28	1.19	1.19
2.5	3.75	2.74	2.13	1.52	1.20	1.13	1.12
2.5	4.00	2.58	2.04	1.42	1.13	1.08	1.07
3.0	0.00	1383.62	842.15	502.90	397.46	374.50	370.40
3.0	0.25	133.61	144.74	171.09	208.54	245.76	281.15
3.0	0.50	37.33	37.41	48.45	75.35	110.95	155.22
3.0	0.75	19.95	17.90	20.16	31.46	50.92	81.22
3.0	1.00	13.52	11.38	11.15	15.74	25.64	43.89
3.0	1.25	10.24	8.32	7.39	9.21	14.26	24.96
3.0	1.50	8.26	6.57	5.47	6.11	8.72	14.97
3.0	1.75	6.94	5.45	4.34	4.45	5.80	9.47
3.0	2.00	6.00	4.67	3.62	3.47	4.15	6.30
3.0	2.25	5.30	4.10	3.11	2.84	3.16	4.41
3.0	2.50	4.76	3.67	2.75	2.41	2.52	3.24
3.0	2.75	4.32	3.32	2.47	2.10	2.09	2.49
3.0	3.00	3.97	3.05	2.26	1.87	1.79	2.00
3.0	3.25	3.67	2.82	2.09	1.69	1.57	1.67
3.0	3.50	3.42	2.62	1.95	1.53	1.41	1.45
3.0	3.75	3.22	2.45	1.84	1.41	1.29	1.29
3.0	4.00	3.04	2.30	1.73	1.31	1.20	1.19
3.5	0.00	12851.0	4106.4	2640.16	2227.34	2157.99	2149.34
3.5	0.25	281.09	381.29	625.78	951.18	1245.90	1502.76
3.5	0.50	53.58	64.72	123.43	267.36	468.68	723.81
3.5	0.75	25.62	25.33	38.68	88.70	182.12	334.40
3.5	1.00	16.65	14.79	17.71	35.97	78.05	160.95
3.5	1.25	12.36	10.37	10.48	17.64	37.15	81.80
3.5	1.50	9.86	8.00	7.25	10.19	19.63	43.96
3.5	1.75	8.22	6.54	5.52	6.70	11.46	24.96
3.5	2.00	7.07	5.55	4.47	4.86	7.33	14.97
3.5	2.25	6.21	4.83	3.77	3.78	5.08	9.47
3.5	2.50	5.55	4.29	3.28	3.10	3.76	6.30
3.5	2.75	5.03	3.87	2.91	2.63	2.94	4.41
3.5	3.00	4.60	3.54	2.63	2.30	2.40	3.24
3.5	3.25	4.25	3.26	2.41	2.05	2.03	2.49
3.5	3.50	3.95	3.03	2.23	1.85	1.76	2.00
3.5	3.75	3.70	2.84	2.10	1.69	1.56	1.67
3.5	4.00	3.47	2.66	1.99	1.55	1.40	1.45

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves the control limit parameters. The following variables can be saved:

Variable	Description
ALPHA	probability (α) of exceeding limits
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LIMITN	sample size associated with the control limits
MEAN	process mean ($\overline{\overline{X}}$ or μ_0)
SIGMAS	multiple (k) of standard error of E_i
STDDEV	process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*subgroup-variable* specified in the EWMACHART statement
TYPE	type (estimate or standard value) of _MEAN_ and _STDDEV_
VAR	*process* specified in the EWMACHART statement
WEIGHT	weight (r) assigned to most recent subgroup mean in computation of EWMA

The OUTLIMITS= data set does not contain the control limits; instead, it contains control limit parameters that can be used to recompute the control limits.

Notes:

1. If the control limits vary with subgroup sample size, the special missing value *V* is assigned to the variable _LIMITN_.
2. If the limits are defined in terms of a multiple k of the standard error of E_i, the value of _ALPHA_ is computed as $\alpha = 2(1 - \Phi(k))$, where $\Phi(\cdot)$ is the standard normal distribution function.
3. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = \Phi^{-1}(1 - \alpha/2)$, where Φ^{-1} is the inverse standard normal distribution function.
4. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the EWMACHART statement.

You can use OUTLIMITS= data sets

- to keep a permanent record of the control limit parameters
- to write reports. You may prefer to use OUTTABLE= data sets for this purpose.
- as LIMITS= data sets in subsequent runs of PROC MACONTROL

For an example of an OUTLIMITS= data set, see "Saving Control Limit Parameters" on page 772.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables can be saved:

- the *subgroup-variable*
- a subgroup mean variable named by *process* suffixed with *X*
- a subgroup standard deviation variable named by *process* suffixed with *S*
- a subgroup EWMA variable named by *process* suffixed with *E*
- a subgroup sample size variable named by *process* suffixed with *N*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the EWMACHART statement. For example, consider the following statements:

```
proc macontrol data=clips;
   ewmachart (gap yldstren)*day / weight    =0.2
                                  outhistory=cliphist;
run;
```

The data set CLIPHIST would contain nine variables named DAY, GAPX, GAPS, GAPE, GAPN, YLDSRENX, YLDSRENS, YLDSRENE, and YLDSRENN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 771.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables can be saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on EWMA chart
EWMA	exponentially weighted moving average
LCLE	lower control limit for EWMA
LIMITN	nominal sample size associated with the control limits
MEAN	process mean
SIGMAS	multiple (k) of the standard error associated with control limits
subgroup	values of the subgroup variable
SUBN	subgroup sample size
SUBS	subgroup standard deviation
SUBX	subgroup mean
UCLE	upper control limit for EWMA
VAR	*process* specified in the EWMACHART statement
WEIGHT	weight (r) assigned to most recent subgroup mean in computation of EWMA

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)
- *symbol-variable*

Notes:

1. Either the variable_ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variables _VAR_ and _EXLIM_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. All other variables are numeric.

For an example of an OUTTABLE= data set, see "Saving Control Limit Parameters" on page 772.

ODS Tables

The following table summarizes the ODS tables that you can request with the EWMACHART statement.

Table 27.21. ODS Tables Produced with the EWMACHART Statement

Table Name	Description	Options
EWMACHART	exponentially weighted moving average chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLEOUT
Parameters	exponentially weighted moving average parameters	TABLE, TABLEALL, TABLEC, TABLEID, TABLEOUT

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC MACONTROL statement. Each *process* specified in the EWMACHART statement must be a SAS variable in the DATA= data set. This variable provides measurements that must be grouped into subgroup samples indexed by the *subgroup-variable*. The *subgroup-variable*, which is specified in the EWMACHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. If the i^{th} subgroup contains n_i items, there should be n_i consecutive observations for which the value of the *subgroup-variable* is the index of the i^{th} subgroup. For example, if each subgroup contains five items and there are 30 subgroup samples, the DATA= data set should contain 150 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the MACONTROL procedure reads all the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 1936).

For an example of a DATA= data set, see "Creating EWMA Charts from Raw Data" on page 766.

LIMITS= Data Set

You can read preestablished control limit parameters from a LIMITS= data set specified in the PROC MACONTROL statement. The LIMITS= data set used by the MACONTROL procedure does not contain the actual control limits, but rather it contains the parameters required to compute the limits. For example, the following statements read parameters from the data set PARMS:*

```
proc macontrol data=parts limits=parms;
   ewmachart gap*day;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the MACONTROL procedure. Such data sets always contain the variables required for a LIMITS= data set; see page 796. The LIMITS= data set can also be created directly using a DATA step.

*In Release 6.09 and earlier releases, it is necessary to specify the READLIMITS option.

When you create a LIMITS= data set, you must provide the variable _WEIGHT_, which specifies the weight parameter used to compute the EWMAs. In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables of length 8.
- The variable _INDEX_ is required if you specify the READINDEX= option. This must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8. Valid values are `ESTIMATE`, `STANDARD`, `STDMEAN`, and `STDSIGMA`.
- BY variables are required if specified with a BY statement.

Some advantages of working with a LIMITS= data set are that

- it facilitates reusing a permanently saved set of parameters
- a distinct set of parameters can be read for each *process* specified in the EWMACHART statement
- it facilitates keeping track of multiple sets of parameters that accumulate for the same *process* as the process evolves over time

For an example, see "Reading Preestablished Control Limit Parameters" on page 775.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC MACONTROL statement. This allows you to reuse OUTHISTORY= data sets that have been created in previous runs of the MACONTROL, SHEWHART, or CUSUM procedures or to read output data sets created with SAS summarization procedures such as PROC MEANS.

A HISTORY= data set used with the EWMACHART statement must contain the following:

- the *subgroup-variable*
- a subgroup mean variable for each *process*
- a subgroup sample size variable for each *process*
- a subgroup standard deviation variable for each *process*

The names of the subgroup mean, subgroup standard deviation, and subgroup sample size variables must be the *process* name concatenated with the suffix characters X, S, and N, respectively.

For example, consider the following statements:

```
proc macontrol history=cliphist;
   ewmachart (gap diameter)*day  / weight=0.2;
run;
```

The data set CLIPHIST must include the variables DAY, GAPX, GAPS, GAPN, DIAMTERX, DIAMTERS, and DIAMTERN.

Although a subgroup EWMA variable (named by the *process* name suffixed with E) is saved in an OUTHISTORY= data set, it is not required in a HISTORY= data set, because the subgroup mean variable is sufficient to compute the EWMAs.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the MACONTROL procedure reads all the observations in a HISTORY= data set. However, if the HISTORY= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 1936 for an example).

For an example of a HISTORY= data set, see "Creating EWMA Charts from Subgroup Summary Data" on page 769.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC MACONTROL statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the MACONTROL procedure.

The following table lists the variables required in a TABLE= data set used with the EWMACHART statement:

Variable	Description
EWMA	exponentially weighted moving average
LCLE	lower control limit for EWMA
LIMITN	nominal sample size associated with the control limits
MEAN	process mean
subgroup-variable	values of the *subgroup-variable*
SUBN	subgroup sample size
SUBS	subgroup standard deviation
SUBX	subgroup mean
UCLE	upper control limit for EWMA
WEIGHT	weight (r) assigned to most recent subgroup mean in computation of EWMA

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable of length 8.

For an example of a TABLE= data set, see "Saving Control Limit Parameters" on page 772.

Methods for Estimating the Standard Deviation

When control limits are computed from the input data, four methods are available for estimating the process standard deviation σ. Three methods (referred to as the default, MVLUE, and RMSDF) are available with subgrouped data. A fourth method is used if the data are individual measurements (see "Default Method for Individual Measurements" on page 804).

Default Method for Subgroup Samples

This method is the default for EWMA charts using subgrouped data. The default estimate of σ is

$$\hat{\sigma} = \frac{s_1/c_4(n_1) + \ldots + s_N/c_4(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, s_i is the sample standard deviation of the i^{th} subgroup

$$s_i = \sqrt{\frac{1}{n_i - 1} \sum_{j=1}^{n_i} (x_{ij} - \bar{X}_i)^2}$$

and

$$c_4(n_i) = \frac{\Gamma(n_i/2)\sqrt{2/(n_i - 1)}}{\Gamma((n_i - 1)/2)}$$

Here $\Gamma(\cdot)$ denotes the gamma function, and $\bar{X}_i$ denotes the i^{th} subgroup mean. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$. If the observations are normally distributed, then the expected value of s_i is $c_4(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the *ASTM Manual on Presentation of Data and Control Chart Analysis* (1976).

MVLUE Method for Subgroup Samples

If you specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). The MVLUE is a weighted average of N unbiased estimates of σ of the form $s_i/c_4(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{h_1 s_1/c_4(n_1) + \ldots + h_N s_N/c_4(n_N)}{h_1 + \ldots + h_N}$$

where

$$h_i = \frac{[c_4(n_i)]^2}{1 - [c_4(n_i)]^2}$$

A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

RMSDF Method for Subgroup Samples

If you specify SMETHOD=RMSDF, a weighted root-mean-square estimate is computed for σ as follows:

$$\hat{\sigma} = \frac{\sqrt{(n_1 - 1)s_1^2 + \cdots + (n_N - 1)s_N^2}}{c_4(n)\sqrt{n_1 + \cdots + n_N - N}}$$

The weights are the degrees of freedom $n_i - 1$. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$.

If the unknown standard deviation σ is constant across subgroups, the root-mean-square estimate is more efficient than the minimum variance linear unbiased estimate.

However, in process control applications it is generally not assumed that σ is constant, and if σ varies across subgroups, the root-mean-square estimate tends to be more inflated than the MVLUE.

Default Method for Individual Measurements

When each subgroup sample contains a single observation ($n_i \equiv 1$), the process standard deviation σ is estimated as

$$\hat{\sigma} = \sqrt{\frac{1}{2(N-1)} \sum_{i=1}^{N-1} (x_{i+1} - x_i)^2}$$

where N is the number of observations, and $x_1, x_2, \ldots, x_N$ are the individual measurements. This formula is given by Wetherill (1977), who states that the estimate of the variance is biased if the measurements are autocorrelated.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical	DATA=	*process*
Vertical	HISTORY=	subgroup mean variable
Vertical	TABLE=	_EWMA_

For example, the following sets of statements specify the label *EWMA of Clip Gaps* for the vertical axis and the label *Day* for the horizontal axis of the EWMA chart:

```
proc macontrol data=clips1;
   ewmachart gap*day / weight=0.3;
   label gap = 'EWMA of Clip Gaps';
   label day = 'Day';
run;

proc macontrol history=cliphist;
   ewmachart gap*day / weight=0.3;
   label gapx = 'EWMA of Clip Gaps';
   label day  = 'Day';
run;

proc macontrol table=cliptab;
   ewmachart gap*day;
   label _ewma_  = 'EWMA of Clip Gaps';
   label day     = 'Day';
run;
```

In this example, the label assignments are in effect only for the duration of the procedure step, and they temporarily override any permanent labels associated with the variables.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples

This section provides advanced examples of the EWMACHART statement.

Example 27.1. Specifying Standard Values for the Process Mean and Process Standard Deviation

See MACEW2 in the SAS/QC Sample Library

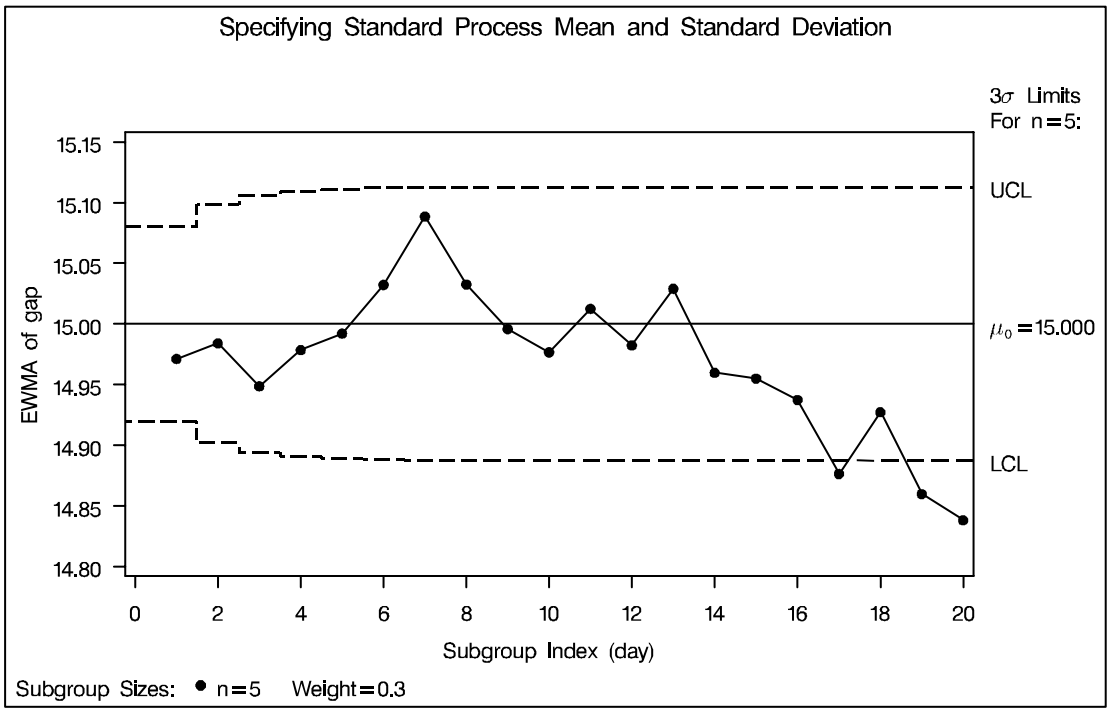
By default, the EWMACHART statement estimates the process mean (μ) and standard deviation (σ) from the data. This is illustrated in the "Getting Started" section of this chapter. However, there are applications in which standard values (μ_0 and σ_0) are available based, for instance, on previous experience or extensive sampling. You can specify these values with the MU0= and SIGMA0= options.

For example, suppose it is known that the metal clip manufacturing process (introduced on page 766) has a mean of 15 and standard deviation of 0.2. The following statements specify these standard values:

```
symbol h = .8;
title 'Specifying Standard Process Mean and Standard Deviation';
symbol v=dot h=.8;
proc macontrol data=clips1;
   ewmachart gap*day /
       mu0     = 15
       sigma0  = 0.2
       weight  = 0.3
       xsymbol = mu0;
run;
```

The XSYMBOL= option specifies the label for the central line. The resulting chart is shown in Output 27.1.1.

Output 27.1.1. Specifying Standard Values with MU0= and SIGMA0=



The central line and control limits are determined using μ_0 and σ_0 (see the equations in Table 27.19 on page 791). Output 27.1.1 indicates that the process is out-of-control, since the moving averages for DAY=17, DAY=19, and DAY=20 lie below the lower control limit.

You can also specify μ_0 and σ_0 with the variables _MEAN_ and _STDDEV_ in a LIMITS= data set, as illustrated by the following statements:

```
data cliplim;
   length _var_ _subgrp_ _type_ $8;
   _var_    = 'gap';
   _subgrp_ = 'day';
   _type_   = 'STANDARD';
   _limitn_ = 5;
   _mean_   = 15;
   _stddev_ = 0.2;
   _weight_ = 0.3;

proc macontrol data=clips1 limits=cliplim;
   ewmachart gap*day / xsymbol=mu0;
run;
```

The variable _WEIGHT_ is required, and its value provides the weight parameter used to compute the EWMAs. The variables _VAR_ and _SUBGRP_ are also required, and their values must match the *process* and *subgroup-variable*, respectively, specified in the EWMACHART statement. The bookkeeping variable _TYPE_ is not required, but it is recommended to indicate that the variables _MEAN_ and _STDDEV_ provide standard values rather than estimated values.

The resulting chart (not shown here) is identical to the one shown in Output 27.1.1.

Example 27.2. Displaying Limits Based on Asymptotic Values

See MACEW3 in the SAS/QC Sample Library

The upper (lower) control limits in Output 27.1.1 are monotonically increasing (decreasing). As the number of subgroups increases, the control limits approach the following asymptotic values:

$$\mathrm{LCL} = \overline{\overline{X}} - k\hat{\sigma}\sqrt{r/n(2-r)}$$

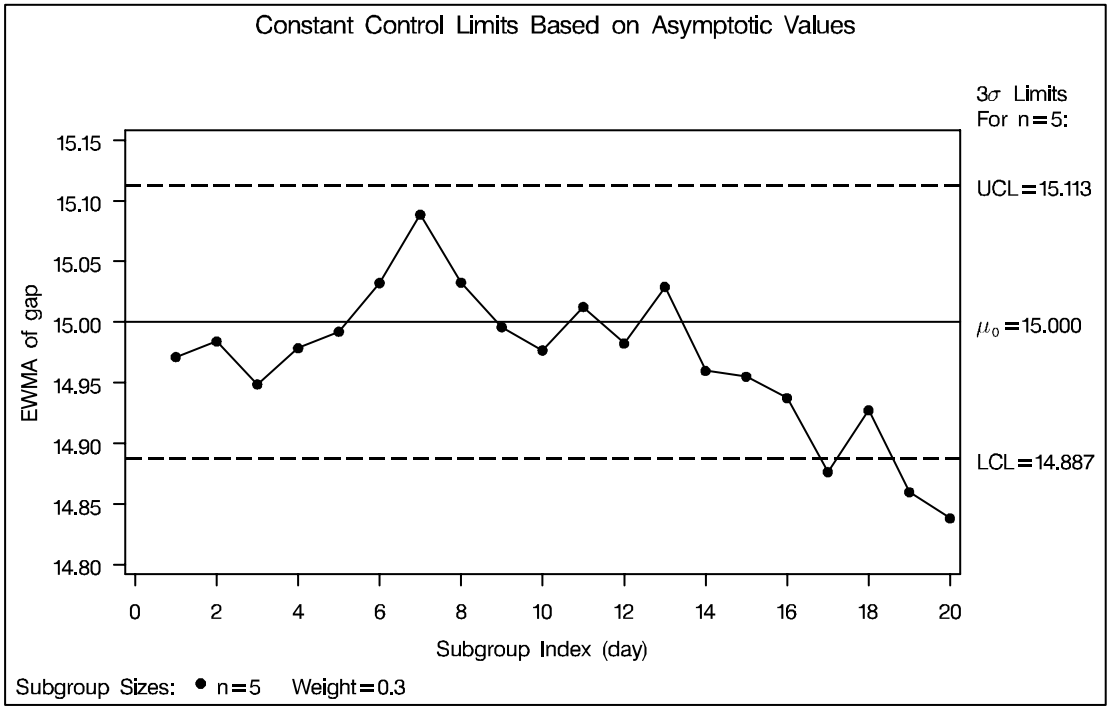
$$\mathrm{UCL} = \overline{\overline{X}} + k\hat{\sigma}\sqrt{r/n(2-r)}$$

These constant limits are displayed if you specify the ASYMPTOTIC option, as illustrated by the following statements:

```
symbol h = .8;
title 'Constant Control Limits Based on Asymptotic Values';
proc macontrol data=clips1;
   ewmachart gap*day /
        mu0     = 15
        sigma0  = 0.2
        weight  = 0.3
        asymptotic
        xsymbol = mu0;
run;
```

The chart is shown in Output 27.2.1.

Output 27.2.1. Asymptotic Control Limits



Note that the same three points that were outside the exact limits (displayed in Output 27.1.1) fall outside the asymptotic limits. The exact limits quickly approach the asymptotic values, so only the first few subgroups have appreciably different limits.

Example 27.3. Working with Unequal Subgroup Sample Sizes

See MACEW4 in the SAS/QC Sample Library

This example contains measurements from the metal clip manufacturing process (introduced on page 766). The following statements create a SAS data set named CLIPS4, which contains additional clip gap measurements taken on a daily basis:

```
data clips4;
   input day @;
   length dayc $2.;
   informat day ddmmyy8.;
   format   day date5.;
   dayc=put(day,date5.);
   dayc=substr(dayc,1,2);
   do i=1 to 5;
      input gap @;
      output;
      end;
   drop i;
   label dayc='April';
datalines;
 1/4/86  14.93  14.65  14.87  15.11  15.18
 2/4/86  15.06  14.95  14.91  15.14  15.41
 3/4/86  14.90  14.90  14.96  15.26  15.18
 4/4/86  15.25  14.57  15.33  15.38  14.89
 7/4/86  14.68  14.63  14.72  15.32  14.86
 8/4/86  14.48  14.88  14.98  14.74  15.48
 9/4/86  14.99  15.16  15.02  15.53  14.66
10/4/86  14.88  15.44  15.04  15.10  14.89
11/4/86  15.14  15.33  14.75  15.23  14.64
14/4/86  15.46  15.30  14.92  14.58  14.68
15/4/86  15.23  14.63    .      .      .
16/4/86  15.13  15.25    .      .      .
17/4/86  15.06  15.25  15.28  15.30  15.34
18/4/86  15.22  14.77  15.12  14.82  15.29
21/4/86  14.95  14.96  14.65  14.87  14.77
22/4/86  15.01  15.11  15.11  14.79  14.88
23/4/86  14.97  15.50  14.93  15.13  15.25
24/4/86  15.23  15.21  15.31  15.07  14.97
25/4/86  15.08  14.75  14.93  15.34  14.98
28/4/86  15.07  14.86  15.42  15.47  15.24
29/4/86  15.27  15.20  14.85  15.62  14.67
30/4/86  14.97  14.73  15.09  14.98  14.46
;
run;
```

Note that only two gap measurements were recorded on April 15 and April 16.

A partial listing of CLIPS4 is shown in Output 27.3.1. This data set contains three variables: DAY is a numeric variable that contains the date (month, day, and year)

that the measurement is taken, DAYC is a character variable that contains the day the measurement is taken, and GAP is a numeric variable that contains the measurement.

Output 27.3.1. The Data Set CLIPS4

The Data Set CLIPS4

day	dayc	gap
01APR	01	14.93
01APR	01	14.65
01APR	01	14.87
01APR	01	15.11
01APR	01	15.18
02APR	02	15.06
02APR	02	14.95
02APR	02	14.91
02APR	02	15.14
02APR	02	15.41
.	.	.
.	.	.
.	.	.
30APR	30	14.46

The following statements request an EWMA chart, shown in Output 27.3.2, for these gap measurements:

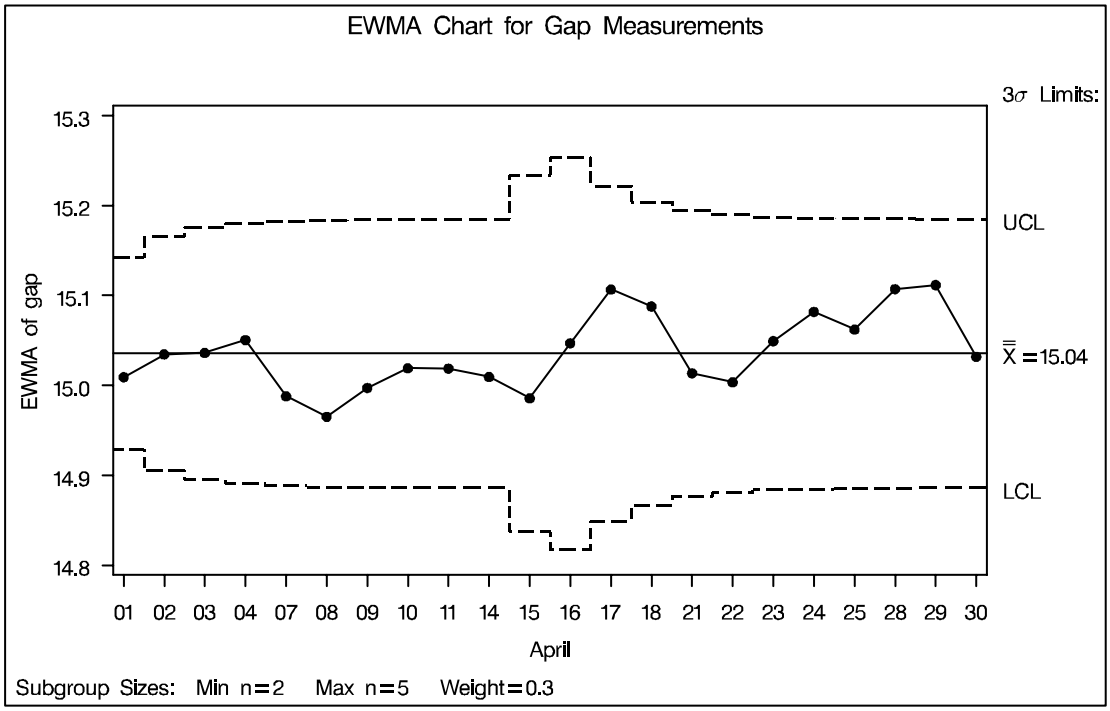
```
symbol h = .8;
title 'EWMA Chart for Gap Measurements';
proc macontrol data=clips4;
   ewmachart gap*dayc / weight = 0.3;
run;
```

The character variable DAYC (rather than the numeric variable DAY) is specified as the *subgroup-variable* in the preceding EWMACHART statement. If DAY were the *subgroup-variable*, each day during April would appear on the horizontal axis, including the weekend days of April 5 and April 6 for which no measurements were taken. To avoid this problem, the *subgroup-variable* DAYC is created from DAY using the PUT and SUBSTR function. Since DAYC is a character *subgroup-variable*, a discrete axis is used for the horizontal axis, and as a result, April 5 and April 6 do not appear on the horizontal axis in Output 27.3.2. A LABEL statement is used to specify the label *April* for the horizontal axis, indicating the month that these measurements were taken.

Note that the control limits vary with the subgroup sample size. The sample size legend in the lower left corner displays the minimum and maximum subgroup sample sizes.

The EWMACHART statement provides various options for working with unequal subgroup sample sizes. For example, you can use the LIMITN= option to specify a fixed (nominal) sample size for computing control limits, as illustrated by the following statements:

Output 27.3.2. EWMA Chart with Varying Sample Sizes




```
symbol h = .8;
title 'EWMA Chart for Gap Measurements';
proc macontrol data=clips4;
   ewmachart gap*dayc / weight = 0.3
                        limitn = 5;
run;
```

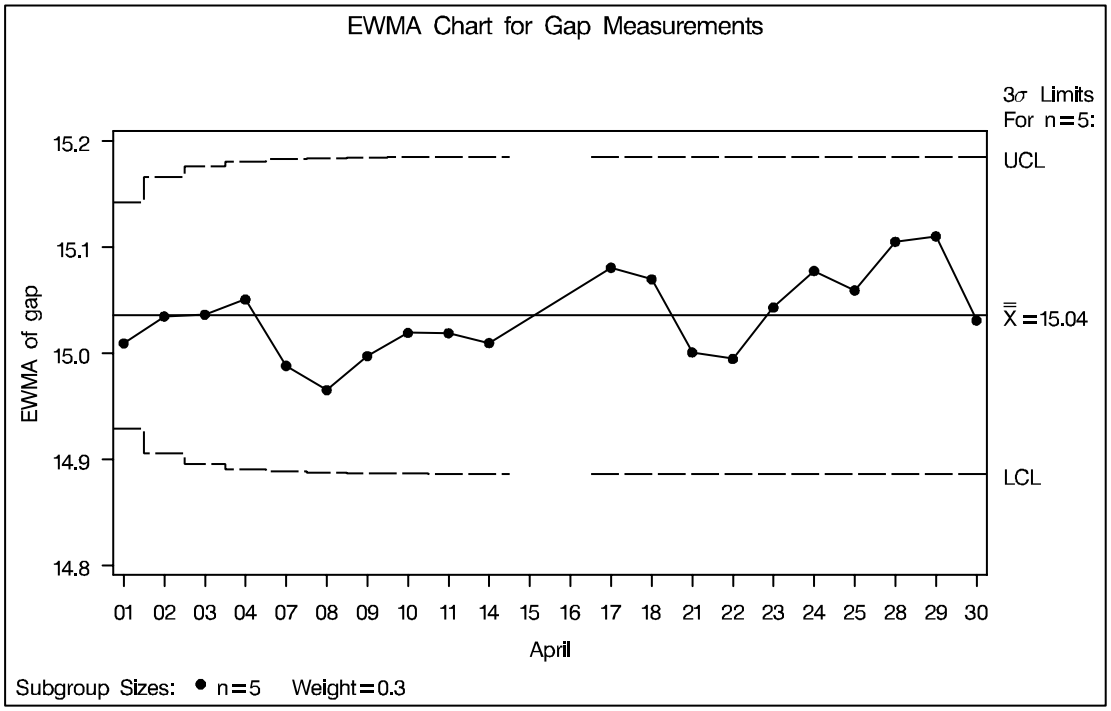
The resulting chart is shown in Output 27.3.3.

Note that the only points displayed are those corresponding to subgroups whose sample size matches the nominal sample size of five. Therefore, points are not displayed for April 15 and April 16. To plot points for all subgroups (regardless of subgroup sample size), you can specify the ALLN option, as follows:

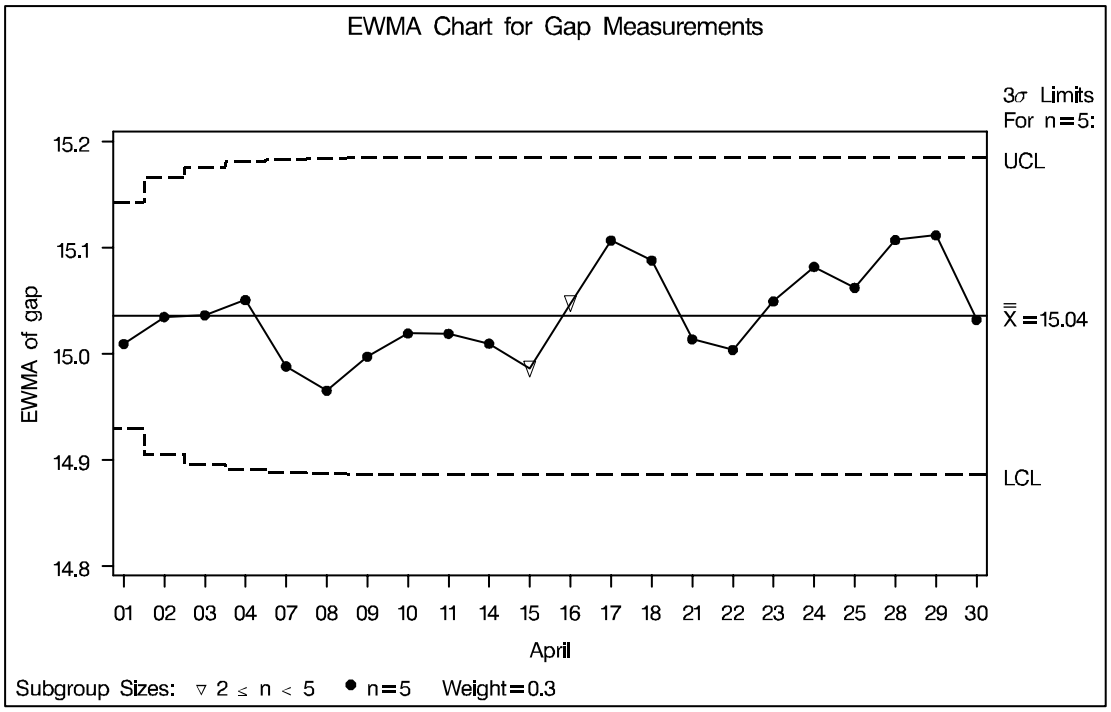
```
symbol h = .8;
title 'EWMA Chart for Gap Measurements';
proc macontrol data=clips4;
   ewmachart gap*dayc/ weight=0.3
                       limitn=5
                       alln
                       nmarkers;
run;
```

The chart is shown in Output 27.3.4. The NMARKERS option requests special symbols to identify points for which the subgroup sample size differs from the nominal sample size.

Output 27.3.3. Control Limits Based on Fixed Sample Size



Output 27.3.4. Displaying All Subgroups Regardless of Sample Size



You can use the SMETHOD= option to determine how the process standard deviation σ is to be estimated when the subgroup sample sizes vary. The default method computes $\hat{\sigma}$ as an unweighted average of subgroup estimates of σ. Specifying SMETHOD=MVLUE requests a minimum variance linear unbiased estimate (MVLUE), which assigns greater weight to estimates of σ from subgroups with larger sample sizes. Specifying SMETHOD=RMSDF requests a weighted root-mean-square estimate. If the unknown standard deviation σ is constant across subgroups, the root-mean-square estimate is more efficient than the MVLUE. For more information, see "Methods for Estimating the Standard Deviation" on page 802.

The following statements apply all three methods:

```
proc macontrol data=clips4;
   ewmachart gap*dayc / outlimits = cliplim1
                        outindex  = 'Default'
                        weight    = 0.3
                        nochart;
   ewmachart gap*dayc / smethod   = mvlue
                        outlimits = cliplim2
                        outindex  = 'MVLUE'
                        weight    = 0.3
                        nochart;
   ewmachart gap*dayc / smethod   = rmsdf
                        outlimits = cliplim3
                        outindex  = 'RMSDF'
                        weight    = 0.3
                        nochart;

run;

data climits;
   set cliplim1 cliplim2 cliplim3;
run;
```

The data set CLIMITS is listed in Output 27.3.5.

Output 27.3.5. Listing of the Data Set CLIMITS

Estimating the Process Standard Deviation

VAR	_SUBGRP_	_INDEX_	_TYPE_	_MIN_	_ALPHA_	_SIGMA_	_MEAN_	_STDDEV_	_WEIGHT_
gap	dayc	Default	ESTIMATE	V	.002699796	3	15.0354	0.26503	0.3
gap	dayc	MVLUE	ESTIMATE	V	.002699796	3	15.0354	0.26096	0.3
gap	dayc	RMSDF	ESTIMATE	V	.002699796	3	15.0354	0.25959	0.3

Note that the estimate of the process standard deviation (stored in the variable _STDDEV_) is slightly different depending on the estimation method. The variable _LIMITN_ is assigned the special missing value *V* in the OUTLIMITS= data set, indicating that the subgroup sample sizes vary.

Example 27.4. Displaying Individual Measurements on an EWMA Chart

In the manufacture of automotive tires, the diameter of the steel belts inside the tire is measured. The following data set contains these measurements for 30 tires:

See MACEW5 in the SAS/QC Sample Library

```
data tires;
   input sample diameter @@;
   datalines;
 1  24.05    2  23.99    3  23.95
 4  23.93    5  23.97    6  24.02
 7  24.06    8  24.10    9  23.98
10  24.03   11  23.91   12  24.06
13  24.06   14  23.96   15  23.98
16  24.06   17  24.01   18  24.00
19  23.93   20  23.92   21  24.09
22  24.11   23  24.05   24  23.98
25  23.98   26  24.06   27  24.02
28  24.06   29  23.97   30  23.96
;
run;
```

The following statements use the IRCHART statement in the SHEWHART procedure (see Chapter 41, "IRCHART Statement,") to create a data set containing the control limits for individual measurements and moving range charts for DIAMETER:

```
proc shewhart data=tires;
   irchart diameter*sample / nochart outlimits=tlimits;
run;
```

A listing of the data set TLIMITS is shown in Output 27.4.1.

Output 27.4.1. Listing of the Data Set TLIMITS

Control Limits for Diameter Measurements

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_
diameter	sample	ESTIMATE	2	.002699796	3

LCLI	_MEAN_	_UCLI_	_LCLR_	_R_	_UCLR_	_STDDEV_
23.8571	24.0083	24.1596	0	0.056897	0.18585	0.050423

The upper and lower control limits for the diameter measurements are 24.1596 and 23.8571, respectively.

In this example, reference lines will be used to display the control limits for the individual measurements on the EWMA chart. The following DATA step reads these control limits from TLIMITS and creates a data set named VREFDATA, which contains the reference line information:

```
data vrefdata;
   set tlimits;
   length _reflab_ $16.;
   keep _ref_ _reflab_;
   _ref_ = _lcli_; _reflab_= 'LCL for X'; output;
   _ref_ = _ucli_; _reflab_= 'UCL for X'; output;
run;
```

A listing of the data set VREFDATA is shown in Output 27.4.2.

Output 27.4.2. Listing of the Data Set VREFDATA

Reference Line Information

reflab	_ref_
LCL for X	23.8571
UCL for X	24.1596

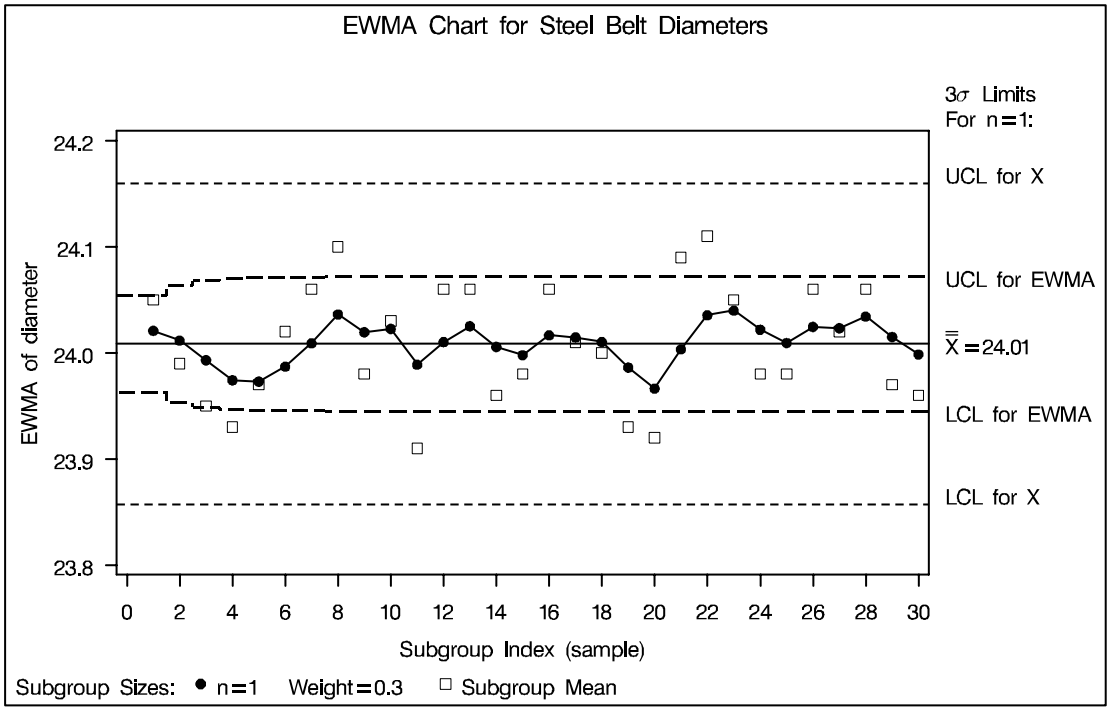
The following statements request an EWMA chart for these measurements:

```
symbol h = .8;
title 'EWMA Chart for Steel Belt Diameters';
proc macontrol data=tires;
   ewmachart diameter*sample / weight     = 0.3
                               meansymbol = square
                               lcllabel   = 'LCL for EWMA'
                               ucllabel   = 'UCL for EWMA'
                               vref       = vrefdata
                               vreflabpos = 3;
run;
```

The MEANSYMBOL= option displays the individual measurements on the EWMA chart. By default, these values are not displayed. The MEANSYMBOL= option specifies the symbol used to plot the individual measurements. The VREF= option reads the reference line information from VREFDATA. The resulting chart is shown in Output 27.4.3.

Output 27.4.3 indicates that the process is in control. None of the diameter measurements (indicated by squares) exceed their control limits, and none of the EWMAs exceed their limits.

Output 27.4.3. Displaying Individual Measurements on EWMA Chart



Example 27.5. Computing Average Run Lengths

See MACEW6 in the SAS/QC Sample Library

The EWMAARL DATA step function computes the average run length for an exponentially weighted moving average (EWMA) scheme (refer to Crowder 1987a,b for details). You can use this function to design a scheme by first calculating average run lengths for a range of values for the weight and then choosing the weight that yields a desired average run length.

The following statements compute the average run lengths for shifts between 0.5 and 2 and weights between 0.25 and 1. The data set ARLS is displayed in Output 27.5.1.

```
data arls;
   do shift=.5 to 2 by .5;
      do weight=.25 to 1 by .25;
      arl=ewmaarl(shift,weight,3.0);
      output;
      end;
   end;
run;

title 'Average Run Lengths for Various Shifts and Weights';
proc print data=arls noobs;
   by shift;
run;
```

Output 27.5.1. Listing of the Data Set ARLS

```
            Average Run Lengths for Various Shifts and Weights

-------------------------------- shift=0.5 --------------------------------

                              weight      arl

                               0.25      48.453
                               0.50      75.354
                               0.75     110.950
                               1.00     155.224


-------------------------------- shift=1 ----------------------------------

                              weight      arl

                               0.25     11.1543
                               0.50     15.7378
                               0.75     25.6391
                               1.00     43.8947


-------------------------------- shift=1.5 --------------------------------

                              weight      arl

                               0.25      5.4697
                               0.50      6.1111
                               0.75      8.7201
                               1.00     14.9677


-------------------------------- shift=2 ----------------------------------

                              weight      arl

                               0.25     3.61677
                               0.50     3.46850
                               0.75     4.15346
                               1.00     6.30296
```

Note that when the weight is 1.0, the EWMAARL function returns the average run length for a Shewhart chart for means. For more details, see "EWMAARL Function" on page 2103.

In addition to using the EWMAARL function to design a EWMA scheme with desired average run length properties, you can use it to evaluate an existing scheme. For example, suppose you have an EWMA chart with 3σ control limits using a weight parameter of 0.3. The following DATA step computes the average run lengths for various shifts using this scheme:

```
data arlinfo;
   do shift=0 to 2 by .25;
      arl = ewmaarl(shift,0.3,3.0);
      output;
   end;
run;
```

```
title 'Average Run Lengths for EWMA Scheme (k=3 and r=0.3)';
proc print data=arlinfo noobs;
run;
```

The data set ARLINFO is displayed in Output 27.5.2.

Output 27.5.2. Listing of the Data Set ARLINFO

Average Run Lengths for EWMA Scheme (k=3 and r=0.3)

shift	arl
0.00	465.553
0.25	178.741
0.50	53.160
0.75	21.826
1.00	11.699
1.25	7.525
1.50	5.447
1.75	4.258
2.00	3.506

Chapter 28
MACHART Statement

Chapter Contents

OVERVIEW 821

GETTING STARTED 822
- Creating Moving Average Charts from Raw Data 822
- Creating Moving Average Charts from Subgroup Summary Data 825
- Saving Summary Statistics 827
- Saving Control Limit Parameters 828
- Reading Preestablished Control Limit Parameters 830

SYNTAX 832
- Summary of Options 834
- Dictionary of Special Options 841

DETAILS 845
- Constructing Uniformly Weighted Moving Average Charts 845
- Output Data Sets 851
- ODS Tables 853
- Input Data Sets 853
- Methods for Estimating the Standard Deviation 857
- Axis Labels 858
- Missing Values 859

EXAMPLES 859
- Example 28.1. Specifying Standard Values for the Process Mean and Process Standard Deviation 859
- Example 28.2. Annotating Average Run Lengths on the Chart 861

Chapter 28
MACHART Statement

Overview

The MACHART statement creates a uniformly weighted moving average control chart (commonly referred to as a moving average control chart), which is used to decide whether a process is in a state of statistical control and to detect shifts in the process average.

You can use options in the MACHART statement to

- specify the span of the moving averages (the number of terms in the moving average)
- compute control limits from the data based on a multiple of the standard error of the plotted moving averages or as probability limits
- tabulate the moving averages, subgroup sample sizes, subgroup means, subgroup standard deviations, control limits, and other information
- save control limit parameters in an output data set
- save the moving averages, subgroup sample sizes, subgroup means, and subgroup standard deviations in an output data set
- read control limit parameters from an input data set
- specify one of several methods for estimating the process standard deviation
- specify a known (standard) process mean and standard deviation for computing control limits
- display a secondary chart that plots a time trend that has been removed from the data
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

Getting Started

This section introduces the MACHART statement with simple examples that illustrate the most commonly used options. Complete syntax for the MACHART statement is presented in the "Syntax" section on page 832, and advanced examples are given in the "Examples" section on page 859.

Creating Moving Average Charts from Raw Data

See MACMA1 in the SAS/QC Sample Library

In the manufacture of a metal clip, the gap between the ends of the clip is a critical dimension. To monitor the process for a change in the average gap, subgroup samples of five clips are selected daily. The data are analyzed with a uniformly weighted moving average chart. The gaps recorded during the first twenty days are saved in a SAS data set named CLIPS1.

```
data clips1;
   input day @ ;
   do i=1 to 5;
      input gap @ ;
      output;
      end;
   drop i;
   datalines;
 1  14.76  14.82  14.88  14.83  15.23
 2  14.95  14.91  15.09  14.99  15.13
 3  14.50  15.05  15.09  14.72  14.97
 4  14.91  14.87  15.46  15.01  14.99
 5  14.73  15.36  14.87  14.91  15.25
 6  15.09  15.19  15.07  15.30  14.98
 7  15.34  15.39  14.82  15.32  15.23
 8  14.80  14.94  15.15  14.69  14.93
 9  14.67  15.08  14.88  15.14  14.78
10  15.27  14.61  15.00  14.84  14.94
11  15.34  14.84  15.32  14.81  15.17
12  14.84  15.00  15.13  14.68  14.91
13  15.40  15.03  15.05  15.03  15.18
14  14.50  14.77  15.22  14.70  14.80
15  14.81  15.01  14.65  15.13  15.12
16  14.82  15.01  14.82  14.83  15.00
17  14.89  14.90  14.60  14.40  14.88
18  14.90  15.29  15.14  15.20  14.70
19  14.77  14.60  14.45  14.78  14.91
20  14.80  14.58  14.69  15.02  14.85
;
run;
```

The following statements produce the listing of the data set CLIPS1 shown in Figure 28.1:

```
title 'The Data Set CLIPS1';
proc print data=clips1 noobs;
run;
```

The Data Set CLIPS1

day	gap
1	14.76
1	14.82
1	14.88
1	14.83
1	15.23
2	14.95
2	14.91
2	15.09
2	14.99
2	15.13
.	.
.	.
.	.
20	14.80
20	14.58
20	14.69
20	15.02
20	14.85

Figure 28.1. Partial Listing of the Data Set CLIPS1

The data set CLIPS1 is said to be in "strung-out" form, since each observation contains the day and gap measurement of a single clip. The first five observations contain the gap measurements for the first day, the second five observations contain the gap measurements for the second day, and so on. Because the variable DAY classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable GAP contains the gap measurements and is referred to as the *process variable* (or *process* for short).

The within-subgroup variability of the gap measurements is known to be stable. You can use a uniformly weighted moving average chart to determine whether the mean level is in control. The following statements create the chart shown in Figure 28.2:

```
symbol h = .8;
title 'Moving Average Chart for Gap Measurements';
proc macontrol data=clips1;
   machart gap*day / span=3;
run;
```

This example illustrates the basic form of the MACHART statement. After the keyword MACHART, you specify the *process* to analyze (in this case, GAP) followed by an asterisk and the *subgroup-variable* (DAY). The SPAN= option specifies the number of terms to include in the moving average. Options such as SPAN= are specified after the slash (/) in the MACHART statement. A complete list of options is presented in the "Syntax" section on page 832. You must provide the span of the

moving average. As an alternative to specifying the SPAN= option, you can read the span from an input data set; see "Reading Preestablished Control Limit Parameters" on page 830.

The input data set is specified with the DATA= option in the PROC MACONTROL statement.

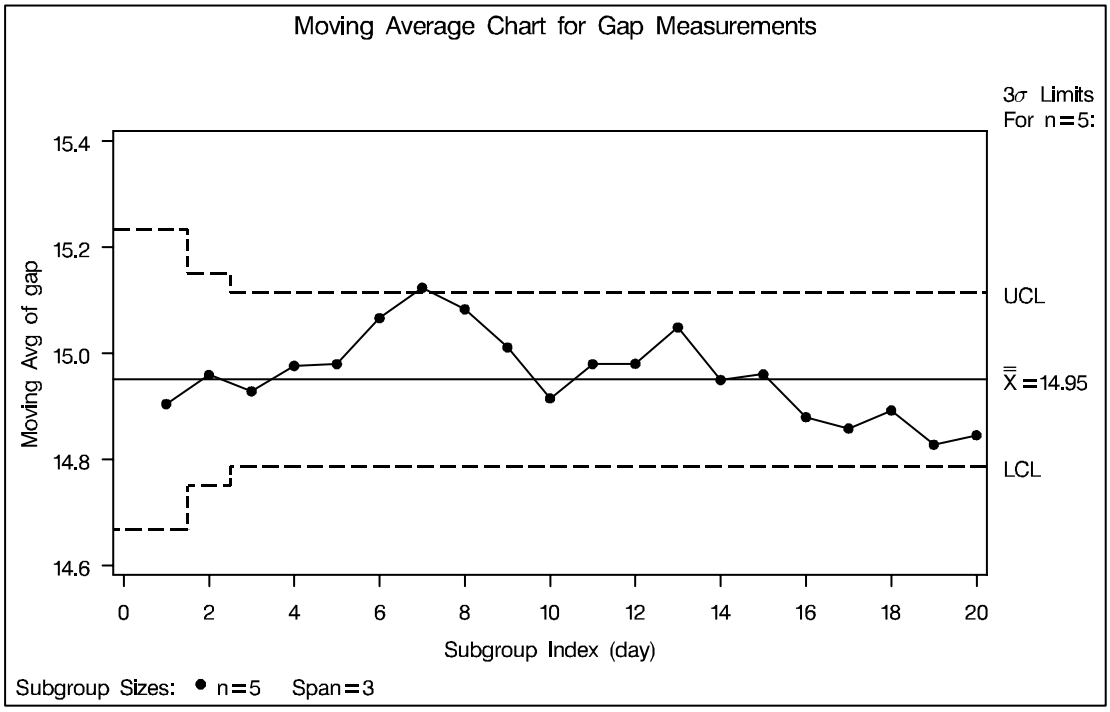


Figure 28.2. Uniformly Weighted Moving Average Chart for Gap Data

Each point on the chart represents the uniformly weighted moving average for a particular day. The moving average A_1 plotted at DAY=1 is simply the subgroup mean for DAY=1. The moving average A_2 plotted at DAY=2 is the average of the subgroup means for DAY=1 and DAY=2. The moving average A_3 plotted at DAY=3 is the average of the subgroup means for DAY=1, DAY=2, and DAY=3.

$$
\begin{aligned}
A_1 &= \frac{14.76 + 14.82 + 14.88 + 14.83 + 15.23}{5} = 14.904 \text{ mm} \\
A_2 &= \frac{14.904 + 15.014}{2} = 14.959 \text{ mm} \\
A_3 &= \frac{14.904 + 15.014 + 14.866}{3} = 14.928 \text{ mm}
\end{aligned}
$$

For succeeding days, the moving average is similarly calculated as the average of the present and the two previous subgroup means (since a span of three is specified with the SPAN= option).

Note that the moving average for the seventh day lies above the upper control limit, signaling an out-of-control process.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 28.19 on page 846.

For computational details, see "Constructing Uniformly Weighted Moving Average Charts" on page 845. For more details on reading from a DATA= data set, see "DATA= Data Set" on page 853.

Creating Moving Average Charts from Subgroup Summary Data

See MACMA1 in the SAS/QC Sample Library

The previous example illustrates how you can create moving average charts using raw data (process measurements). However, in many applications the data are provided as subgroup summary statistics. This example illustrates how you can use the MACHART statement with data of this type. The following data set (CLIPSUM) provides the data from the preceding example in summarized form:

```
data clipsum;
   input day gapx gaps;
   gapn=5;
datalines;
 1  14.904  0.18716
 2  15.014  0.09317
 3  14.866  0.25006
 4  15.048  0.23732
 5  15.024  0.26792
 6  15.126  0.12260
 7  15.220  0.23098
 8  14.902  0.17254
 9  14.910  0.19824
10  14.932  0.24035
11  15.096  0.25618
12  14.912  0.16903
13  15.138  0.15928
14  14.798  0.26329
15  14.944  0.20876
16  14.896  0.09965
17  14.734  0.22512
18  15.046  0.24141
19  14.702  0.17880
20  14.788  0.16634
;
run;
```

A partial listing of CLIPSUM is shown in Figure 28.3. There is exactly one observation for each subgroup (note that the subgroups are still indexed by DAY). The variable GAPX contains the subgroup means, the variable GAPS contains the subgroup standard deviations, and the variable GAPN contains the subgroup sample sizes (these are all five).

The Data Set CLIPSUM

day	gapx	gaps	gapn
1	14.904	0.18716	5
2	15.014	0.09317	5
3	14.866	0.25006	5
.	.	.	.
.	.	.	.
.	.	.	.
20	14.788	0.16634	5

Figure 28.3. The Summary Data Set CLIPSUM

You can read this data set by specifying it as a HISTORY= data set in the PROC MACONTROL statement, as follows:

```
title 'Moving Average Chart for Gap Measurements';
proc macontrol history=clipsum lineprinter;
   machart gap*day='*' / span=3;
run;
```

The resulting moving average chart is shown in Figure 28.4. Since the LINEPRINTER option is specified in the PROC MACONTROL statement, line printer output is produced. The asterisk (*) specified in single quotes after the *subgroup-variable* indicates the character used to plot points. This character must follow an equal sign.

Note that GAP is *not* the name of a SAS variable in the data set but is, instead, the common prefix for the names of the three SAS variables GAPX, GAPS, and GAPN. The suffix characters *X*, *S*, and *N* indicate *mean*, *standard deviation*, and *sample size*, respectively. Thus, you can specify three subgroup summary variables in a HISTORY= data set with a single name (GAP), which is referred to as the *process*. The variables GAPX, GAPS, and GAPN are all required. The name DAY specified after the asterisk is the name of the *subgroup-variable*.

In general, a HISTORY= input data set used with the MACHART statement must contain the following variables:

- subgroup variable
- subgroup mean variable
- subgroup standard deviation variable
- subgroup sample size variable

Furthermore, the names of subgroup mean, standard deviation, and sample size variables must begin with the *process* name specified in the MACHART statement and end with the special suffix characters *X*, *S*, and *N*, respectively. If the names do not follow this convention, you can use the RENAME option in the PROC MACONTROL statement to rename the variables for the duration of the MACONTROL procedure step (see page 1743 for an example).

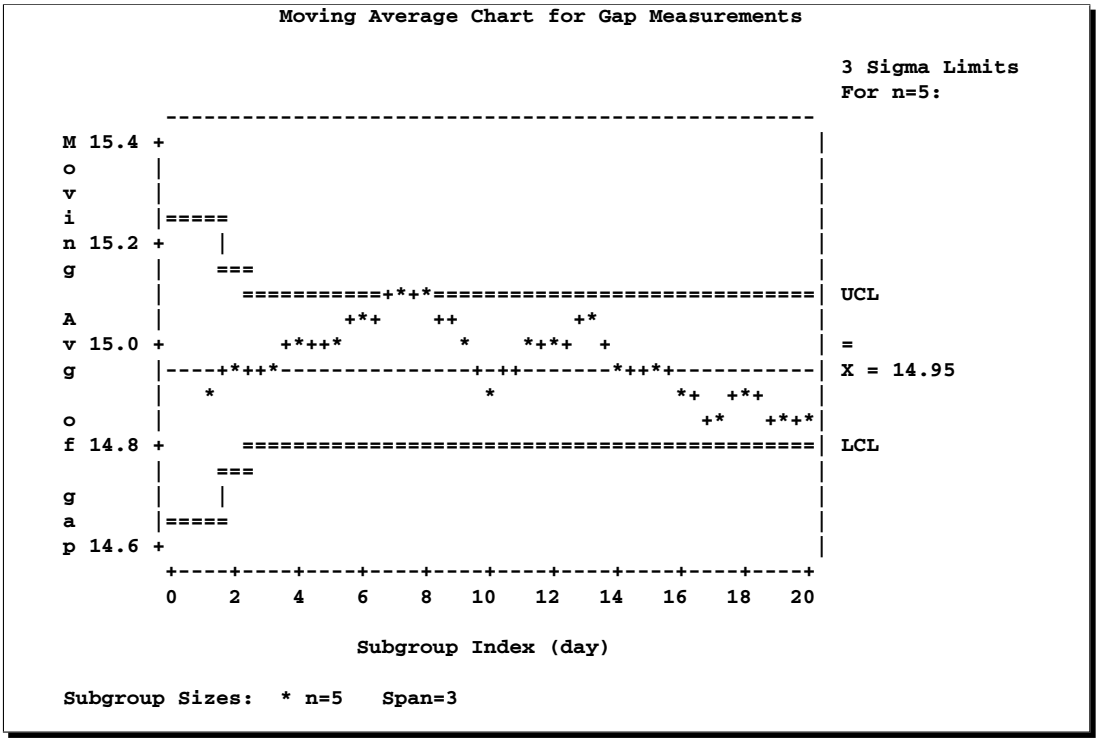


Figure 28.4. Uniformly Weighted Moving Average Chart from Summary Data

In summary, the interpretation of *process* depends on the input data set.

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names of the variables containing the summary statistics.

For more information, see "HISTORY= Data Set" on page 855.

Saving Summary Statistics

In this example, the MACHART statement is used to create a summary data set that can be read later by the MACONTROL procedure (as in the preceding example). The following statements read measurements from the data set CLIPS1 and create a summary data set named CLIPHIST:

See MACMA1 in the SAS/QC Sample Library

```
title 'Summary Data Set for Gap Measurements';
proc macontrol data=clips1;
   machart gap*day / span       = 3
                     outhistory = cliphist
                     nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 28.2.

Figure 28.5 contains a partial listing of CLIPHIST.

Summary Data Set for Gap Measurements

day	gapX	gapS	gapA	gapN
1	14.904	0.18716	14.9040	5
2	15.014	0.09317	14.9590	5
3	14.866	0.25006	14.9280	5
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
20	14.788	0.16634	14.8453	5

Figure 28.5. The Summary Data Set CLIPHIST

There are five variables in the data set CLIPHIST.

- DAY contains the subgroup index.
- GAPX contains the subgroup means.
- GAPS contains the subgroup standard deviations.
- GAPA contains the subgroup moving averages.
- GAPN contains the subgroup sample sizes.

Note that the summary statistic variables are named by adding the suffix characters *X*, *S*, *A*, and *N* to the *process* GAP specified in the MACHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 852.

Saving Control Limit Parameters

See MACMA1 in the SAS/QC Sample Library

You can save the control limit parameters used for a moving average chart in a SAS data set; this enables you to use these parameters with future data (see "Reading Preestablished Control Limit Parameters" on page 830) or modify the parameters with a DATA step program.

The following statements read measurements from the data set CLIPS1 (see page 822) and save the control limit parameters in a data set named CLIPLIM:

```
title 'Control Limit Parameters';
proc macontrol data=clips1;
   machart gap*day / span      = 3
                     outlimits = cliplim
                     nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. The data set CLIPLIM is listed in Figure 28.6.

Control Limit Parameters

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_MEAN_	_STDDEV_	_SPAN_
gap	day	ESTIMATE	5	.002699796	3	14.95	0.21108	3

Figure 28.6. The Data Set CLIPLIM Containing Control Limit Information

Note that the data set CLIPLIM does not contain the actual control limits, but rather the parameters required to compute the limits.

The data set contains one observation with the parameters for *process* GAP. The variable _SPAN_ contains the number of terms used to calculate the moving average. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates that the values of _MEAN_ and _STDDEV_ are estimates rather than standard values. For more information, see "OUTLIMITS= Data Set" on page 851.

You can create an output data set containing the control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
title 'Summary Statistics and Control Limits';
proc macontrol data=clips1;
   machart gap*day / span     = 3
                     outtable = cliptab
                     nochart;
run;
```

The data set CLIPTAB is listed in Figure 28.7.

This data set contains one observation for each subgroup sample. The variable _UWMA_ contains the uniformly weighted moving average. The variables _SUBX_, _SUBS_, and _SUBN_ contain the subgroup means, subgroup standard deviations, and subgroup sample sizes, respectively. The variables _LCLA_ and _UCLA_ contain the lower and upper control limits, and the variable _MEAN_ contains the central line. The variables _VAR_ and DAY contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 852.

Summary Statistics and Control Limits

VAR	day	_SIGMA_	_LIMITS_	_SUBN_	_SUBX_	_SUBS_	_CLA_	_UWA_	_MEAN_	_UCL_	_EXLIM_
gap	1	3	5	3	5	14.904	0.18716	14.6668	14.9040	14.95	15.2332	
gap	2	3	5	3	5	15.014	0.09317	14.7498	14.9590	14.95	15.1502	
gap	3	3	5	3	5	14.866	0.25006	14.7865	14.9280	14.95	15.1135	
gap	4	3	5	3	5	15.048	0.23732	14.7865	14.9760	14.95	15.1135	
gap	5	3	5	3	5	15.024	0.26792	14.7865	14.9793	14.95	15.1135	
gap	6	3	5	3	5	15.126	0.12260	14.7865	15.0660	14.95	15.1135	
gap	7	3	5	3	5	15.220	0.23098	14.7865	15.1233	14.95	15.1135	UPPER
gap	8	3	5	3	5	14.902	0.17254	14.7865	15.0827	14.95	15.1135	
gap	9	3	5	3	5	14.910	0.19824	14.7865	15.0107	14.95	15.1135	
gap	10	3	5	3	5	14.932	0.24035	14.7865	14.9147	14.95	15.1135	
gap	11	3	5	3	5	15.096	0.25618	14.7865	14.9793	14.95	15.1135	
gap	12	3	5	3	5	14.912	0.16903	14.7865	14.9800	14.95	15.1135	
gap	13	3	5	3	5	15.138	0.15928	14.7865	15.0487	14.95	15.1135	
gap	14	3	5	3	5	14.798	0.26329	14.7865	14.9493	14.95	15.1135	
gap	15	3	5	3	5	14.944	0.20876	14.7865	14.9600	14.95	15.1135	
gap	16	3	5	3	5	14.896	0.09965	14.7865	14.8793	14.95	15.1135	
gap	17	3	5	3	5	14.734	0.22512	14.7865	14.8580	14.95	15.1135	
gap	18	3	5	3	5	15.046	0.24141	14.7865	14.8920	14.95	15.1135	
gap	19	3	5	3	5	14.702	0.17880	14.7865	14.8273	14.95	15.1135	
gap	20	3	5	3	5	14.788	0.16634	14.7865	14.8453	14.95	15.1135	

Figure 28.7. The OUTTABLE= Data Set CLIPTAB

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read CLIPTAB and display a moving average chart (not shown here) identical to Figure 28.2:

```
title 'Moving Average Chart for Gap Measurements';
proc macontrol table=cliptab;
   machart gap*day;
run;
```

For more information, see "TABLE= Data Set" on page 856.

Reading Preestablished Control Limit Parameters

See MACMA1 in the SAS/QC Sample Library

In the previous example, the OUTLIMITS= data set saved the control limit parameters in the data set CLIPLIM. This example shows how to apply these parameters to new data provided in the following data set:

```
data clips1a;
   label gap='Gap Measurement (mm)';
   input day @;
   do i=1 to 5;
      input gap @;
      output;
   end;
```

```
   drop i;
datalines;
21  14.86 15.01 14.67 14.67 15.07
22  14.93 14.53 15.07 15.10 14.98
23  15.27 14.90 15.12 15.10 14.80
24  15.02 15.21 14.93 15.11 15.20
25  14.90 14.81 15.26 14.57 14.94
26  14.78 15.29 15.13 14.62 14.54
27  14.78 15.15 14.61 14.92 15.07
28  14.92 15.31 14.82 14.74 15.26
29  15.11 15.04 14.61 15.09 14.68
30  15.00 15.04 14.36 15.20 14.65
31  14.99 14.76 15.18 15.04 14.82
32  14.90 14.78 15.19 15.06 15.06
33  14.95 15.10 14.86 15.27 15.22
34  15.03 14.71 14.75 14.99 15.02
35  15.38 14.94 14.68 14.77 14.83
36  14.95 15.43 14.87 14.90 15.34
37  15.18 14.94 15.32 14.74 15.29
38  14.91 15.15 15.06 14.78 15.42
39  15.34 15.34 15.41 15.36 14.96
40  15.12 14.75 15.05 14.70 14.74
;
run;
```

The following statements create a moving average chart for the data in CLIPS1A using the control limit parameters in CLIPLIM:

```
symbol h = .8;
title 'Moving Average Chart for Second Set of Gap Measurements';
proc macontrol data=clips1a limits=cliplim;
   machart gap*day;
run;
```

The chart is shown in Figure 28.8.

The LIMITS= option in the PROC MACONTROL statement specifies the data set containing the control limits parameters. By default,* this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name GAP
- the value of _SUBGRP_ matches the *subgroup-variable* name DAY

Note that the moving average plotted for the 39^{th} day lies above the upper control limit, signalling an out-of-control process.

In this example, the LIMITS= data set was created in a previous run of the MACONTROL procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 854 for details concerning the variables that you must provide, and see Example 28.1 on page 859 for an illustration.

*In Release 6.09 and in earlier releases, it is also necessary to specify the READLIMITS option to read control limits from a LIMITS= data set.

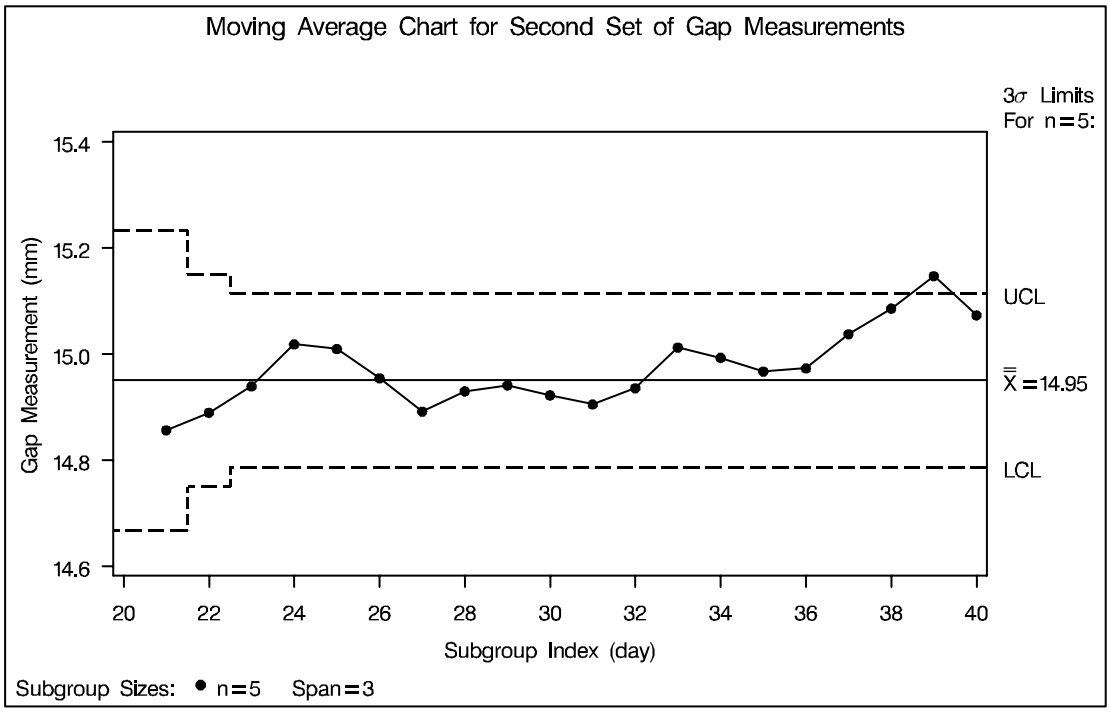


Figure 28.8. Using Control Limit Parameters from a LIMITS= Data Set

Syntax

The basic syntax for the MACHART statement is as follows:

MACHART *process*subgroup-variable* **/ SPAN=***value* < *options* > **;**

The general form of this syntax is as follows:

MACHART (*processes***)****subgroup-variable* <**(** *block-variables* **)** >
< =*symbol-variable* | =’*character*’ > **/ SPAN=***value* < *options* > **;**

Note that the SPAN= option is required unless its *value* is read from a LIMITS= data set. You can use any number of MACHART statements in the MACONTROL procedure. The components of the MACHART statement are described as follows.

process
processes
identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC MACONTROL statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see “Creating Moving Average Charts from Raw Data” on page 822.

- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating Moving Average Charts from Subgroup Summary Data" on page 825.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limit Parameters" on page 828.

A *process* is required. If more than one *process* is specified, enclose the list in parentheses. For example, the following statements request distinct moving average charts (each with a span of 3) for WEIGHT, LENGTH, and WIDTH:

```
proc macontrol data=measures;
   machart (weight length width)*day / span=3;
run;
```

subgroup-variable
is the variable that classifies the data into subgroups. The *subgroup-variable* is required. In the preceding MACHART statement, DAY is the subgroup variable. For details, see "Subgroup Variables" on page 1771.

block-variables
are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 1932 for an example.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker or plotting character used to plot the moving averages.

- If you produce a chart on a line printer, an 'A' is displayed for points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for points corresponding to the second level, and so on.
- If you produce a chart on a graphics device, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOLn statements. See "Displaying Stratification in Levels of a Classification Variable" on page 1931 for an example.

character
specifies a plotting character for charts produced on line printers. For example, the following statements create a moving average chart using an asterisk (*) to plot the points:

```
proc macontrol data=values;
   machart weight*hour='*' / span=3;
run;
```

options

specify chart parameters, enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The "Summary of Options" section, which follows, lists all options by function.

Summary of Options

The following tables list the MACHART statement options by function. Options unique to the MACONTROL procedure are listed in Table 28.1 and Table 28.2, and they are described in detail in "Dictionary of Special Options" on page 841. Options that are common to both the MACONTROL and SHEWHART procedures are listed in Table 28.3 to Table 28.18. They are described in detail beginning on page 1851 .

Table 28.1. Options for Specifying Uniformly Weighted Moving Average Charts

ALPHA=*value*	requests probability limits for control charts
ASYMPTOTIC	requests constant control limits
LIMITN=n|VARYING	specifies either a fixed nominal sample size (n) for control limits or allows the control limits to vary with subgroup sample size
MU0=*value*	specifies a standard (known) value μ_0 for the process mean
NOREADLIMITS	specifies that control limit parameters are not to be read from LIMITS= data set (Release 6.10 and later releases)
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from LIMITS= data set when both variables are available
READINDEX='*value*'	reads control limit parameters from the first observation in the LIMITS= data set where the variable _INDEX_ equals *value*
READLIMITS	reads control limit parameters from a LIMITS= data set (Release 6.09 and earlier releases)
SIGMA0=*value*	specifies standard (known) value σ_0 for process standard deviation
SIGMAS=k	specifies width of control limits in terms of multiple k of standard error of plotted moving averages
SPAN=*value*	specifies the number of terms in the moving average

Table 28.2. Options for Plotting Subgroup Means

CMEANSYMBOL=*color*	specifies color for MEANSYMBOL= symbol
MEANCHAR='*character*'	specifies *character* to plot subgroup means on line printer
MEANSYMBOL= *keyword*	specifies symbol to plot subgroup means on graphics device

Table 28.3. Tabulation Options

TABLE	creates a basic table of subgroup variable values, subgroup sample sizes, subgroup means, subgroup moving averages, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, and TABLEOUTLIM
TABLECENTRAL	augments basic table with the value of the central line
TABLEID	augments basic table with columns for ID variables
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded

Note that specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points.

Table 28.4. Axis and Axis Label Options

CAXIS=*color*	specifies color for axis lines and tick marks
CFRAME=*color*|(*color-list*)	specifies fill colors for frame for plot area
CTEXT=*color*	specifies color for tick mark values and axis labels
HAXIS=*values*|AXISn	specifies major tick mark values for horizontal axis
HEIGHT=*value*	specifies height of axis label and axis legend text
HMINOR=*n*	specifies minor tick marks between major horizontal tick marks
HOFFSET=*value*	specifies length of offset at both ends of horizontal axis
INTSTART=*value*	specifies first major tick mark value for numeric horizontal axis
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=*n*	specifies thinning factor for tick mark labels on horizontal axis
SPLIT='*character*'	specifies splitting character for axis labels
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=*values*|AXISn	specifies major tick mark values for vertical axis on moving average chart
VAXIS2=*values*|AXISn	specifies major tick mark values for vertical axis on trend chart
VFORMAT=*format*	specifies format for tick mark labels on vertical axis of moving average chart
VFORMAT2=*format*	specifies format for tick mark labels on vertical axis of trend chart
VMINOR=*n*	specifies minor tick marks between major vertical tick marks
VOFFSET=*value*	specifies length of offset at both ends of vertical axis
WAXIS=*n*	specifies width of axis lines

Table 28.5. Process Mean and Standard Deviation Options

SMETHOD=*keyword*	specifies method for estimating process standard deviation σ
TYPE=*keyword*	identifies whether parameters are estimates or standard values and specifies value of _TYPE_ in OUTLIMITS= data set

Table 28.6. Grid Options

CGRID=*color*	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to chart
LENDGRID=*linetype*	specifies line type for grid requested with the ENDGRID option
LGRID=*linetype*	specifies line type for grid requested with the GRID option
WGRID=*n*	specifies width of grid lines

Table 28.7. Reference Line Options

CHREF=*color*	specifies color for HREF= and HREF2= lines
CVREF=*color*	specifies color for VREF= and VREF2= lines
HREF=*values*| *SAS-data-set*	specifies reference lines perpendicular to horizontal axis on moving average chart
HREF2=*values*| *SAS-data-set*	specifies reference lines perpendicular to horizontal axis on trend chart
HREFCHAR='*character*'	specifies line character for HREF= and HREF2= lines
HREFDATA= *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on moving average chart
HREF2DATA= *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFLABELS= *'label1'...'labeln'*	specifies labels for HREF= lines
HREF2LABELS= *'label1'...'labeln'*	specifies labels for HREF2= lines
HREFLABPOS=*n*	specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=*linetype*	specifies line type for HREF= and HREF2= lines
LVREF=*linetype*	specifies line type for VREF= and VREF2= lines
NOBYREF	specifies that reference line information in a data set is to be applied uniformly to charts created for all BY groups
VREF=*values*| *SAS-data-set*	specifies reference lines perpendicular to vertical axis on moving average chart
VREF2=*values*| *SAS-data-set*	specifies reference lines perpendicular to vertical axis on trend chart
VREFCHAR='*character*'	specifies line character for VREF= and VREF2= lines
VREFLABELS= *'label1'...'labeln'*	specifies labels for VREF= lines
VREF2LABELS= *'label1'...'labeln'*	specifies labels for VREF2= lines
VREFLABPOS=*n*	specifies position of VREFLABELS= and VREF2LABELS= labels

Table 28.8. Block Variable Legend Options

BLOCKLABELPOS=*keyword*	specifies position of label for *block-variable* legend
BLOCKLABTYPE=*value*|*keyword*	specifies text size of *block-variable* legend
BLOCKPOS=*n*	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=*color-list*	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=*variable*|(*variables*)	specifies one or more variables whose values are colors for filling background of *block-variable* legend

Table 28.9. Options for Displaying Control Limits

CINFILL=*color*	specifies color for area inside control limits
CLIMITS=*color*	specifies color of control limits, central line, and related labels
LCLLABEL='*label*'	specifies label for lower control limit
LIMLABSUBCHAR='*character*'	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=*linetype*	line type for control limits
NDECIMAL=*n*	specifies number of digits to right of decimal place in default labels for control limits and central line
NOCTL	suppresses display of central line
NOLCL	suppresses display of lower control limit
NOLIMITLABEL	suppresses labels for control limits and center line
NOLIMITS	suppresses display of control limits
NOLIMITSLEGEND	suppresses legend for control limits
NOUCL	suppresses display of upper control limit
UCLLABEL='*string*'	specifies label for upper control limit
WLIMITS=*n*	width for control limits and central line
XSYMBOL='*string*'|*keyword*	specifies label for central line

Table 28.10. Options for Interactive Control Charts

HTML=(*variable*)	specifies a variable whose values are URLs to be associated with subgroups
HTML2=(*variable*)	specifies variable whose values are URLs to be associated with points on trend chart
HTML_LEGEND=(*variable*)	specifies a variable whose values are URLs to be associated with symbols in the symbol legend
WEBOUT=*SAS-data-set*	creates an OUTTABLE= data set with additional graphics co-ordinate data

Table 28.11. Options for Plotting and Labeling Points

ALLLABEL=VALUE|(*variable*)	labels every point on moving average chart
ALLLABEL2=VALUE|(*variable*)	labels every point on trend chart
CLABEL=*color*	specifies color for labels
CCONNECT=*color*	specifies color for line segments that connect points on chart
CFRAMELAB=*color*	specifies fill color for frame around labeled points
CNEEDLES=*color*	specifies color for needles that connect points to central line
CONNECTCHAR='*character*'	specifies character used to form line segments that connect points on moving average chart
COUT=*color*	specifies color for line segments that connect points exceeding control limits
COUTFILL=*color*	specifies color for areas between connected points and control limits
LABELANGLE=*angle*	specifies angle at which labels are drawn
LABELFONT=*font*	specifies a software font for labels requested by the ALLLABEL=, ALLLABEL2=, OUTLABEL=, and STARLABEL= options
LABELHEIGHT=*font*	specifies the height (in vertical percent screen units) for labels requested by the ALLLABEL=, ALLLABEL2=, OUTLABEL=, and STARLABEL= options
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on moving average chart
NOTRENDCONNECT	suppresses line segments that connect points on trend chart
OUTLABEL=VALUE|(*variable*)	labels points exceeding control limits
SYMBOLCHARS='*characters*'	specifies characters indicating *symbol-variable*
SYMBOLLEGEND=NONE|*name*	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=*keyword*	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL	turns point labels so that they are strung out vertically

Table 28.12. Input Data Set Options

MISSBREAK	specifies that observations with missing values are not to be processed

Table 28.13. Output Data Set Options

OUTHISTORY= *SAS-data-set*	creates output data set containing subgroup summary statistics
OUTINDEX='*string*'	specifies value of the variable _INDEX_ in OUTLIMITS= data set
OUTLIMITS= *SAS-data-set*	creates output data set containing control limit parameters
OUTPHASE='*string*'	specifies value of the variable _PHASE_ in OUTHISTORY= or OUTTABLE= data set
OUTTABLE= *SAS-data-set*	creates output data set containing subgroup summary statistics and control limits

Table 28.14. Plot Layout Options

ALLN	plots moving averages for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts only when exceptions occur
INTERVAL=*keyword*	specifies natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=*n*	specifies maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of moving average chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=*n*	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=*n*	specifies number of pages or screens to be used to display chart
TRENDVAR=*variable*| (*variable-list*)	specifies list of trend variables
YPCT1=*value*	specifies length of vertical axis on EWMA chart as a percentage of sum of lengths of vertical axes for EWMA and trend charts
ZEROSTD	displays $\bar{X}$ chart regardless of whether $\hat{\sigma} = 0$

Table 28.15. Phase Options

CPHASELEG=*color*	specifies text color for *phase* legend
OUTPHASE='*string*'	specifies value of _PHASE_ in OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=*value*| *keyword*	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASEREF	delineates *phases* with vertical reference lines
READPHASES= ALL| '*label1*'...'*labeln*'	specifies *phases* to be read from input data set

Table 28.16. Graphical Enhancement Options

ANNOTATE= *SAS-data-set*	specifies annotate data set that adds features to moving average chart
ANNOTATE2= *SAS-data-set*	specifies annotate data set that adds features to trend chart
DESCRIPTION='*string*'	specifies string that appears in the description field of PROC GREPLAY master menu for moving average chart
FONT=*font*	specifies software font for labels and legends on chart
NAME='*string*'	specifies name that appears in the name field of the PROC GREPLAY master menu for moving average chart
PAGENUM='*string*'	specifies the form of the label used in pagination
PAGENUMPOS= *keyword*	specifies the position of the page number requested with the PAGENUM= option

Table 28.17. Clipping Options

CCLIP=*color*	color for plot symbol for clipped points
CLIPCHAR='*character*'	plot character for clipped points
CLIPFACTOR=*value*	determines extent to which extreme points are clipped
CLIPLEGEND='*string*'	text for clipping legend
CLIPLEGPOS=*keyword*	position of clipping legend
CLIPSUBCHAR= '*character*'	substitution character for CLIPLEGEND= text
CLIPSYMBOL=*symbol*	plot symbol for clipped points

Table 28.18. Star Options

CSTARCIRCLES=*color*	specifies color for STARCIRCLES= circles
CSTARFILL=*color*| *(variable)*	specifies color for filling stars
CSTAROUT=*color*	specifies outline color for stars exceeding inner or outer circles
CSTARS=*color*| *(variable)*	specifies color for outlines of stars
LSTARCIRCLES= *linetypes*	specifies line types for STARCIRCLES= circles
LSTARS=*linetype*| *(variable)*	specifies line types for outlines of stars requested with the STARVERTICES= option
STARBDRADIUS=*value*	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=*value-list*	specifies reference circles for stars
STARINRADIUS=*value*	specifies inner radius of stars
STARLABEL=*keyword*	specifies vertices to be labeled
STARLEGEND=*keyword*	specifies style of legend for star vertices
STARLEGENDLAB= *'label'*	specifies label for STARLEGEND= legend
STAROUTRADIUS=*value*	specifies outer radius of stars
STARSPEC=*value*| *SAS-data-set*	specifies method used to standardize vertex variables
STARSTART=*value*	specifies angle for first vertex
STARTYPE=*keyword*	specifies graphical style of star
STARVERTICES= *variable*|*(variables)*	superimposes star at each point on moving average chart
WSTARCIRCLES=*n*	specifies width of STARCIRCLES= circles
WSTARS=*n*	specifies width of STARVERTICES= stars

Dictionary of Special Options

The marginal notes *Graphics* and *Line Printer* identify options that apply to graphics devices and line printers, respectively.

ALPHA=*value*

requests *probability limits*. If you specify ALPHA=α, the control limits are computed so that the probability is α that a single moving average exceeds its control limits. The value of α can range between 0 and 1. This assumes that the process is in statistical control and that the data follow a normal distribution. For the equations used to compute probability limits, see "Control Limits" on page 845.

Note the following:

- As an alternative to specifying ALPHA=α, you can read α from the variable _ALPHA_ in a LIMITS= data set by specifying the READALPHA option.
- As an alternative to specifying ALPHA=α (or reading _ALPHA_ from a LIMITS= data set), you can request "$k\sigma$ control limits" by specifying SIGMAS=k (or reading _SIGMAS_ from a LIMITS= data set).

If you specify neither the ALPHA= option nor the SIGMAS= option, the procedure computes 3σ control limits by default.

ASYMPTOTIC
requests constant upper and lower control limits for all subgroups having the following values:

$$\begin{aligned} \text{LCL} &= \overline{\overline{X}} - \frac{k\hat{\sigma}}{\sqrt{nw}} \\ \text{UCL} &= \overline{\overline{X}} + \frac{k\hat{\sigma}}{\sqrt{nw}} \end{aligned}$$

Here w is the span of the moving average, and n is the nominal sample size associated with the control limits. Substitute $\Phi^{-1}(1-\alpha/2)$ for k if you specify probability limits with the ALPHA= option. When you do not specify the ASYMPTOTIC option, the control limits are computed using the exact formulas in Table 28.19 on page 846. Use the ASYMPTOTIC option only if all the subgroup sample sizes are the same or if you specify LIMITN=n.

CMEANSYMBOL=*color*
Graphics
specifies the *color* for the symbol requested with the MEANSYMBOL= option. The default *color* is the first color in the device color list.

LIMITN=*n*
LIMITN=VARYING
specifies either a fixed or varying nominal sample size for the control limits.

If you specify LIMITN=n, moving averages are calculated and displayed only for those subgroups with a sample size equal to n, unless you also specify the ALLN option, which causes all the moving averages to be calculated and displayed. By default (or if you specify LIMITN=VARYING), moving averages are calculated and displayed for all subgroups, regardless of sample size.

MEANCHAR='*character***'**
Line Printer
specifies a *character* used to plot the subgroup mean for each subgroup. By default, subgroup means are not plotted.

MEANSYMBOL=*keyword*
Graphics
specifies a symbol used to plot the subgroup mean for each subgroup. By default, subgroup means are not plotted.

MU0=*value*
specifies a known (standard) value μ_0 for the process mean μ. By default, μ is estimated from the data.

Note: As an alternative to specifying MU0=μ_0, you can read a predetermined value for μ_0 from the variable _MEAN_ in a LIMITS= data set.

See Example 28.1 on page 859.

NOREADLIMITS
specifies that control limit parameters for each *process* listed in the MACHART

statement are *not* to be read from the LIMITS= data set specified in the PROC MACONTROL statement. The NOREADLIMITS option is available only in Release 6.10 and later releases.

The following example illustrates the NOREADLIMITS option:

```
proc macontrol data=pistons limits=diamlim;
   machart diameter*hour;
   machart diameter*hour / noreadlimits span=3;
run;
```

The first MACHART statement reads the control limits from the first observation in the data set DIAMLIM for which the variable _VAR_ is equal to `diameter` and the variable _SUBGRP_ is equal to `hour`. The second MACHART statement computes estimates of the process mean and standard deviation for the control limits from the measurements in the data set PISTONS. Note that the second MACHART statement is equivalent to the following statements, which would be more commonly used:

```
proc macontrol data=pistons;
   machart diameter*hour / span=3;
run;
```

For more information about reading control limit parameters from a LIMITS= data set, see the READLIMITS option later in this list.

READALPHA
specifies that the variable _ALPHA_, rather than the variable _SIGMAS_, is to be read from a LIMITS= data set when both variables are available in the data set. Thus the limits displayed are probability limits. If you do not specify the READALPHA option, then _SIGMAS_ is read by default.

READINDEX=*'value'*
reads control limit parameters from a LIMITS= data set (specified in the PROC MACONTROL statement) for each *process* listed in the MACHART statement. The control limit parameters for a particular *process* are read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches *process*
- the value of _SUBGRP_ matches the *subgroup-variable*
- the value of _INDEX_ matches *value*

The *value* can be up to 48 characters and must be enclosed in quotes.

READLIMITS
specifies that control limit parameters are to be read from a LIMITS= data set specified in the PROC MACONTROL statement. The parameters for a particular *process* are read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches *process*
- the value of _SUBGRP_ matches the *subgroup variable*

The use of the READLIMITS option depends on which release of SAS/QC software you are using.

- **In Release 6.10 and later releases, the READLIMITS option is not necessary.** To read control limits parameters as described previously, you simply specify a LIMITS= data set. However, even though the READLIMITS option is redundant, it continues to function as in earlier releases.
- **In Release 6.09 and earlier releases, you must specify the READLIMITS option to read control limits parameters as described previously.** If you specify a LIMITS= data set without specifying the READLIMITS option (or the READINDEX= option), the control limits are computed from the data and the span of the moving averages is specified with the SPAN= option.

SIGMA0=*value*
specifies a known (standard) value σ_0 for the process standard deviation σ. The *value* must be positive. By default, the MACONTROL procedure estimates σ from the data using the formulas given in "Methods for Estimating the Standard Deviation" on page 857.

Note: As an alternative to specifying SIGMA0=σ_0, you can read a predetermined value for σ_0 from the variable _STDDEV_ in a LIMITS= data set.

SIGMAS=*value*
specifies the width of the control limits in terms of the multiple k of the standard error of the plotted moving averages on the chart. The value of k must be positive. By default, $k = 3$ and the control limits are 3σ limits.

SPAN=*value*
specifies the number of terms used to calculate the moving average (*value* is an integer greater than 1). The SPAN= option is required unless you read control limit parameters from a LIMITS= data set or a TABLE= data set. See "Plotted Points" on page 845 and "Choosing the Span of the Moving Average" on page 847 for details.

Details

Constructing Uniformly Weighted Moving Average Charts

The following notation is used in this section:

A_i	uniformly weighted moving average for the i^{th} subgroup
w	span parameter (number of terms in moving average)
μ	process mean (expected value of the population of measurements)
σ	process standard deviation (standard deviation of the population of measurements)
x_{ij}	j^{th} measurement in i^{th} subgroup, with j =1, 2, 3, ..., n_i
n_i	sample size of i^{th} subgroup
$\overline{X}_i$	mean of measurements in i^{th} subgroup. If $n_i = 1$, then the subgroup mean reduces to the single observation in the subgroup.
$\overline{\overline{X}}$	weighted average of subgroup means
$\Phi^{-1}(\cdot)$	inverse standard normal function

Plotted Points

Each point on the chart indicates the value of the uniformly weighted moving average for that subgroup. The moving average for the i^{th} subgroup (A_i) is defined as

$$A_i = (\overline{X}_1 + \ldots + \overline{X}_i)/i \qquad \text{if } i < w$$

$$A_i = (\overline{X}_i + \ldots + \overline{X}_{i-w+1})/w \quad \text{if } i \geq w$$

where w is the span, or number of terms, of the moving average. You can specify the span with the SPAN= option in the MACHART statement or with the value of _SPAN_ in a LIMITS= data set.

Central Line

By default, the central line on a moving average chart indicates an estimate for μ, which is computed as

$$\hat{\mu} = \overline{\overline{X}} = \frac{n_1\bar{X}_1 + \cdots + n_N\bar{X}_N}{n_1 + \cdots + n_N}$$

If you specify a known value (μ_0) for μ, the central line indicates the value of μ_0.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of A_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).

- as probability limits defined in terms of α, a specified probability that A_i exceeds the limits

The following table presents the formulas for the limits:

Table 28.19. Limits for Moving Average Chart

Control Limits
LCL = $\overline{\overline{X}} - k(\hat{\sigma}/\min(i,w))\sqrt{(1/n_i) + (1/n_{i-1}) + \ldots + (1/n_{1+\max(i-w,0)})}$
UCL = $\overline{\overline{X}} + k(\hat{\sigma}/\min(i,w))\sqrt{(1/n_i) + (1/n_{i-1}) + \ldots + (1/n_{1+\max(i-w,0)})}$

Probability Limits
LCL = $\overline{\overline{X}} - \Phi^{-1}(1-\alpha/2)(\hat{\sigma}/\min(i,w))\sqrt{(1/n_i) + (1/n_{i-1}) + \ldots + (1/n_{1+\max(i-w,0)})}$
UCL = $\overline{\overline{X}} + \Phi^{-1}(1-\alpha/2)(\hat{\sigma}/\min(i,w))\sqrt{(1/n_i) + (1/n_{i-1}) + \ldots + (1/n_{1+\max(i-w,0)})}$

These formulas assume that the data are normally distributed. If standard values μ_0 and σ_0 are available for μ and σ, respectively, replace $\overline{\overline{X}}$ with μ_0 and replace $\hat{\sigma}$ with σ_0 in Table 28.19. Note that the limits vary with both n_i and i.

If the subgroup sample sizes are constant ($n_i = n$), the formulas for the control limits simplify to

$$\begin{aligned} \text{LCL} &= \overline{\overline{X}} - \frac{k\hat{\sigma}}{\sqrt{n\min(i,w)}} \\ \text{UCL} &= \overline{\overline{X}} + \frac{k\hat{\sigma}}{\sqrt{n\min(i,w)}} \end{aligned}$$

Refer to Montgomery (1996) for more details. When the subgroup sample sizes are constant, the width of the control limits for the first w moving averages decreases monotonically because each of the first w moving averages includes one more term than the preceding moving average.

If you specify the ASYMPTOTIC option, constant control limits with the following values are displayed:

$$\begin{aligned} \text{LCL} &= \overline{\overline{X}} - \frac{k\hat{\sigma}}{\sqrt{nw}} \\ \text{UCL} &= \overline{\overline{X}} + \frac{k\hat{\sigma}}{\sqrt{nw}} \end{aligned}$$

For asymptotic probability limits, replace k with $\Phi^{-1}(1-\alpha/2)$ in these equations. You can display asymptotic limits by specifying the ASYMPTOTIC option.

You can specify parameters for the moving average limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify w with the SPAN= option or with the variable _SPAN_ in a LIMITS= data set.
- Specify μ_0 with the MU0= option or with the variable _MEAN_ in a LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in a LIMITS= data set.

Choosing the Span of the Moving Average

There are few published guidelines for choosing the span w. In some applications, practical experience may dictate the choice of w. A more systematic approach is to choose w by considering its effect on the average run length (the expected number of points plotted before a shift is detected). This effect was studied by Roberts (1959), who used simulation methods.

You can use Table 28.20 and Table 28.21 to find a combination of k and w that yields a desired ARL for an in-control process ($\delta = 0$) and for a specified shift of δ.

Table 28.20. Average Run Lengths for One-Sided Uniformly Weighted Moving Average Charts

		w (span)						
k	δ	2	3	4	5	6	8	10
2.0	0.00	51.58	60.97	70.58	80.18	89.78	108.65	127.47
2.0	0.25	25.01	26.47	28.00	29.33	30.76	33.08	35.18
2.0	0.50	13.41	13.31	13.40	13.69	14.01	14.66	15.17
2.0	0.75	8.00	7.75	7.78	7.97	8.15	8.60	9.06
2.0	1.00	5.27	5.20	5.29	5.45	5.67	6.15	6.69
2.0	1.50	2.90	3.03	3.24	3.50	3.73	4.23	4.66
2.0	2.00	2.04	2.27	2.51	2.73	2.95	3.32	3.65
2.0	2.50	1.68	1.91	2.11	2.31	2.48	2.78	3.04
2.0	3.00	1.46	1.68	1.85	2.01	2.16	2.40	2.63
2.0	4.00	1.20	1.38	1.52	1.64	1.75	1.94	2.10
2.0	5.00	1.06	1.18	1.31	1.41	1.50	1.65	1.79
2.5	0.00	179.92	204.43	230.32	259.32	287.08	339.71	394.43
2.5	0.25	72.62	71.56	72.48	72.93	73.40	75.54	77.47
2.5	0.50	33.67	30.13	28.54	27.49	26.93	26.29	26.03
2.5	0.75	17.28	15.01	13.91	13.42	13.13	13.00	13.10
2.5	1.00	9.94	8.66	8.20	8.01	7.96	8.24	8.63

Table 28.20. (continued)

k	δ	2	3	4	5	6	8	10
2.5	1.50	4.43	4.13	4.21	4.39	4.64	5.17	5.69
2.5	2.00	2.65	2.77	3.03	3.29	3.54	4.01	4.43
2.5	2.50	1.98	2.24	2.50	2.74	2.95	3.32	3.67
2.5	3.00	1.70	1.95	2.17	2.37	2.55	2.86	3.14
2.5	4.00	1.37	1.59	1.76	1.90	2.03	2.28	2.49
2.5	5.00	1.15	1.35	1.51	1.62	1.73	1.92	2.08
3.0	0.00	792.24	867.57	963.95	1051.77	1150.79	1345.96	1539.75
3.0	0.25	269.28	244.26	231.50	226.25	220.89	209.87	204.74
3.0	0.50	104.18	83.86	72.84	65.43	60.85	54.62	50.34
3.0	0.75	45.69	34.45	28.79	25.69	23.66	21.24	20.15
3.0	1.00	22.73	16.74	14.20	12.89	12.12	11.52	11.45
3.0	1.50	7.65	6.16	5.70	5.64	5.75	6.23	6.78
3.0	2.00	3.77	3.49	3.63	3.89	4.17	4.71	5.20
3.0	2.50	2.46	2.63	2.90	3.18	3.43	3.88	4.28
3.0	3.00	1.96	2.23	2.50	2.74	2.95	3.33	3.65
3.0	4.00	1.57	1.81	2.00	2.18	2.34	2.62	2.87
3.0	5.00	1.30	1.55	1.72	1.85	1.97	2.20	2.40
3.5	0.00	4275.15	4536.99	4853.63	5168.75	5485.97	6088.03	6613.01
3.5	0.25	1281.12	1078.59	964.86	886.26	830.03	751.66	684.98
3.5	0.50	413.30	294.47	235.00	197.27	169.50	136.01	115.48
3.5	0.75	153.50	98.31	73.49	59.29	50.49	40.45	34.53
3.5	1.00	63.68	39.34	29.37	24.06	20.88	17.70	16.12
3.5	1.50	15.84	10.44	8.50	7.78	7.47	7.51	7.97
3.5	2.00	6.06	4.73	4.49	4.61	4.86	5.43	6.01
3.5	2.50	3.27	3.13	3.34	3.63	3.92	4.45	4.91
3.5	3.00	2.31	2.54	2.83	3.11	3.36	3.80	4.19
3.5	4.00	1.77	2.02	2.25	2.45	2.64	2.97	3.27
3.5	5.00	1.48	1.74	1.91	2.06	2.21	2.48	2.71

Table 28.21. Average Run Lengths for Two-Sided Uniformly Weighted Moving Average Charts

		w (span)						
k	δ	2	3	4	5	6	8	10
2.0	0.00	25.46	29.62	33.94	38.08	42.35	51.20	59.48
2.0	0.25	20.43	22.38	24.21	25.87	27.35	30.08	32.33
2.0	0.50	12.73	12.80	13.02	13.29	13.57	14.19	14.84
2.0	0.75	7.87	7.68	7.71	7.86	8.03	8.44	8.90
2.0	1.00	5.24	5.14	5.22	5.40	5.59	6.09	6.60
2.0	1.50	2.90	3.02	3.24	3.48	3.71	4.19	4.63
2.0	2.00	2.04	2.26	2.51	2.73	2.94	3.31	3.63
2.0	2.50	1.67	1.91	2.12	2.30	2.47	2.77	3.03

Table 28.21. (continued)

k	δ	2	3	4	5	6	8	10
2.0	3.00	1.46	1.67	1.85	2.01	2.15	2.40	2.63
2.0	4.00	1.20	1.38	1.52	1.64	1.75	1.94	2.10
2.0	5.00	1.06	1.19	1.31	1.41	1.50	1.65	1.79
2.5	0.00	89.48	101.24	114.35	127.74	140.88	166.98	192.93
2.5	0.25	63.12	64.91	67.00	68.75	69.84	72.22	74.49
2.5	0.50	32.46	29.54	28.20	27.33	26.72	25.92	25.72
2.5	0.75	17.28	14.97	13.85	13.29	13.02	12.81	12.98
2.5	1.00	9.94	8.61	8.16	7.99	8.01	8.23	8.63
2.5	1.50	4.42	4.14	4.20	4.38	4.62	5.16	5.67
2.5	2.00	2.65	2.77	3.03	3.29	3.54	4.00	4.43
2.5	2.50	1.99	2.24	2.50	2.73	2.95	3.33	3.65
2.5	3.00	1.69	1.95	2.17	2.37	2.54	2.86	3.14
2.5	4.00	1.37	1.59	1.76	1.90	2.04	2.27	2.49
2.5	5.00	1.15	1.35	1.51	1.63	1.73	1.92	2.09
3.0	0.00	397.12	436.27	481.16	527.14	574.05	667.68	762.89
3.0	0.25	245.51	228.67	222.75	216.07	213.79	207.03	201.71
3.0	0.50	103.15	83.49	72.47	65.67	60.67	53.93	50.30
3.0	0.75	45.56	34.25	29.01	25.72	23.59	21.12	19.93
3.0	1.00	22.68	16.81	14.19	12.92	12.18	11.54	11.48
3.0	1.50	7.68	6.14	5.71	5.65	5.77	6.23	6.77
3.0	2.00	3.74	3.49	3.63	3.88	4.17	4.71	5.21
3.0	2.50	2.46	2.63	2.90	3.18	3.43	3.89	4.29
3.0	3.00	1.96	2.23	2.50	2.73	2.95	3.32	3.66
3.0	4.00	1.57	1.81	2.00	2.18	2.34	2.62	2.88
3.0	5.00	1.30	1.55	1.72	1.85	1.97	2.20	2.40
3.5	0.00	2217.61	2372.09	2567.27	2775.06	2983.70	3398.08	3810.50
3.5	0.25	1186.27	1027.67	940.30	875.91	826.53	744.59	676.61
3.5	0.50	411.69	295.62	232.68	195.65	169.21	135.73	116.06
3.5	0.75	152.52	97.33	72.30	58.98	50.59	40.22	34.71
3.5	1.00	64.03	39.46	29.18	24.08	20.80	17.54	16.16
3.5	1.50	15.83	10.36	8.47	7.73	7.46	7.56	8.00
3.5	2.00	6.05	4.71	4.49	4.61	4.85	5.44	6.00
3.5	2.50	3.27	3.12	3.34	3.64	3.92	4.44	4.91
3.5	3.00	2.32	2.54	2.83	3.11	3.36	3.80	4.19
3.5	4.00	1.77	2.02	2.25	2.46	2.65	2.97	3.26
3.5	5.00	1.49	1.74	1.91	2.06	2.21	2.48	2.71

For example, suppose you want to construct a two-sided moving average chart with an in-control ARL of 100 and an ARL of 9 for detecting a shift of $\delta = 1$. Table 28.21 shows that the combination $w = 3$ and $k = 2.5$ yields an in-control ARL of 101.24 and an ARL of 8.61 for $\delta = 1$.

Note that you can also use Table 28.20 and Table 28.21 to evaluate an existing moving average chart (see Example 28.2 on page 861).

The following SAS program computes the average run length for a two-sided moving average chart for various shifts in the mean. This program can be adapted to compute averages run lengths for various combinations of k and w.

```
data sim;
   drop span delta time j y x;
   span=4;
   do shift=0,.25,.5,.75,1,1.5,2,2.5,3,4,5;
      do j=1 to 50000;
         do time=1 to 15000;
            if time<=100 then
               delta=0;
            else
               delta=shift;
            y=delta+rannor(234);
            if time<span then
               x=.;
            else
               x=(y+lag1(y)+lag2(y)+lag3(y))/span;
            if time>=101 and abs(x)>3/sqrt(span)
              then leave;
            end;
         arl=time-100;
         output;
         end;
      end;

proc means;
   class shift;
run;
```

In the preceding program, the size of the span w (SPAN) is 4 and the shifts in the mean are introduced to the variable (Y) $y \sim N(0,1)$ after the first 100 observations. The first DO loop specifies shifts of various magnitude, the second DO loop performs 50000 simulations for each shift, and the third DO loop counts the run length (TIME), that is, the number of samples observed before the control chart signals. A large upper bound (15000) for TIME is specified so that the run length is uncensored.

The program can be generalized for various span sizes by assigning a different value for the variable SPAN and changing the expression for X appropriately. Optionally, you can compute the ARL for a one-sided chart by changing the limits, that is, **x>3/sqrt(span)**. This was the technique used to construct Table 28.20 and Table 28.21.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves the control limit parameters. The following variables can be saved:

Variable	Description
ALPHA	probability (α) of exceeding limits
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LIMITN	sample size associated with the control limits
MEAN	process mean ($\overline{\overline{X}}$ or μ_0)
SIGMAS	multiple (k) of standard error of A_i
SPAN	number of terms in the moving average
STDDEV	process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*subgroup-variable* specified in the MACHART statement
TYPE	type (estimate or standard value) of _MEAN_ and _STDDEV_
VAR	*process* specified in the MACHART statement

The OUTLIMITS= data set does not contain the control limits; instead, it contains control limit parameters that can be used to recompute the control limits.

Notes:

1. If the control limits vary with subgroup sample size, the special missing value *V* is assigned to the variable _LIMITN_.
2. If the limits are defined in terms of a multiple k of the standard error of A_i, the value of _ALPHA_ is computed as $\alpha = 2(1 - \Phi(k))$, where $\Phi(\cdot)$ is the standard normal distribution function.
3. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = \Phi^{-1}(1 - \alpha/2)$, where Φ^{-1} is the inverse standard normal distribution function.
4. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the MACHART statement.

You can use OUTLIMITS= data sets

- to keep a permanent record of the control limit parameters
- to write reports. You may prefer to use OUTTABLE= data sets for this purpose.
- as LIMITS= data sets in subsequent runs of PROC MACONTROL

For an example of an OUTLIMITS= data set, see "Saving Control Limit Parameters" on page 828.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables can be saved:

- the *subgroup-variable*
- a subgroup mean variable named by *process* suffixed with *X*
- a subgroup standard deviation variable named by *process* suffixed with *S*
- a subgroup moving average variable named by *process* suffixed with *A*
- a subgroup sample size variable named by *process* suffixed with *N*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the MACHART statement. For example, consider the following statements:

```
proc macontrol data=clips;
   machart (gap yldstren)*day / span      =3
                                outhistory=cliphist;
run;
```

The data set CLIPHIST would contain nine variables named DAY, GAPX, GAPS, GAPA, GAPN, YLDSRENX, YLDSRENS, YLDSRENA, and YLDSRENN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 827.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables can be saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on moving average chart
LCLA	lower control limit for moving average
LIMITN	nominal sample size associated with the control limits
MEAN	process mean
SIGMAS	multiple (k) of the standard error associated with control limits
SPAN	number of terms in the moving average
subgroup	values of the subgroup variable
SUBN	subgroup sample size
SUBS	subgroup standard deviation
SUBX	subgroup mean
UCLA	upper control limit for moving average
UWMA	uniformly weighted moving average
VAR	*process* specified in MACHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)
- *symbol-variable*

Notes:

1. Either the variable_ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively; or with the corresponding variables in a LIMITS= data set).
2. The variables _VAR_ and _EXLIM_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. All other variables are numeric.

For an example of an OUTTABLE= data set, see "Saving Control Limit Parameters" on page 828.

ODS Tables

The following table summarizes the ODS tables that you can request with the MACHART statement.

Table 28.22. ODS Tables Produced with the MACHART Statement

Table Name	Description	Options
MACHART	uniformly weighted moving average chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLEOUT
Parameters	uniformly weighted moving average parameters	TABLE, TABLEALL, TABLEC, TABLEID, TABLEOUT

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC MACONTROL statement. Each *process* specified in the MACHART statement must be a SAS variable in the DATA= data set. This variable provides measurements that must be grouped into subgroup samples indexed by the *subgroup-variable*. The *subgroup-variable*, which is specified in the MACHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. If the i^{th} subgroup contains n_i items, there should be n_i consecutive observations for which the value of the *subgroup-variable* is the index of the i^{th} subgroup. For example, if each subgroup contains five items and there are 30 subgroup samples, the DATA= data set should contain 150 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the MACONTROL procedure reads all of the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 1936).

For an example of a DATA= data set, see "Creating Moving Average Charts from Raw Data" on page 822.

LIMITS= Data Set

You can read preestablished control limits parameters from a LIMITS= data set specified in the PROC MACONTROL statement. The LIMITS= data set used by the MACONTROL procedure does not contain the actual control limits, but rather it contains the parameters required to compute the limits. For example, the following statements read control limit parameters from the data set PARMS:*

```
proc macontrol data=parts limits=parms;
   machart gap*day;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the MACONTROL procedure. Such data sets always contain the variables required for a LIMITS= data set; see page 851. The LIMITS= data set can also be created directly using a DATA step.

When you create a LIMITS= data set, you must provide the variable _SPAN_, which specifies the number of terms to use in the moving average. In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables of length 8.
- The variable _INDEX_ is required if you specify the READINDEX= option. This must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8. Valid values are `ESTIMATE`, `STANDARD`, `STDMEAN`, and `STDSIGMA`.
- BY variables are required if specified with a BY statement.

Some advantages of working with a LIMITS= data set are that

- it facilitates reusing a permanently saved set of parameters
- a distinct set of parameters can be read for each *process* specified in the MACHART statement

*In Release 6.09 and earlier releases, it is necessary to specify the READLIMITS option.

- it facilitates keeping track of multiple sets of parameters that accumulate for the same *process* as the process evolves over time

For an example, see "Reading Preestablished Control Limit Parameters" on page 830.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC MACONTROL statement. This allows you to reuse OUTHISTORY= data sets that have been created in previous runs of the MACONTROL, SHEWHART, or CUSUM procedures or to read output data sets created with SAS summarization procedures such as PROC MEANS.

A HISTORY= data set used with the MACHART statement must contain the following:

- the *subgroup-variable*
- a subgroup mean variable for each *process*
- a subgroup sample size variable for each *process*
- a subgroup standard deviation variable for each *process*

The names of the subgroup mean, subgroup standard deviation, and subgroup sample size variables must be the *process* name concatenated with the suffix characters X, S, and N, respectively.

For example, consider the following statements:

```
proc macontrol history=cliphist;
   machart (gap diameter)*day  / span=3;
run;
```

The data set CLIPHIST must include the variables DAY, GAPX, GAPS, GAPN, DIAMTERX, DIAMTERS, and DIAMTERN.

Although a moving average variable (named by the *process* name suffixed with A) is saved in an OUTHISTORY= data set, it is not required in a HISTORY= data set, because the subgroup mean variable is sufficient to compute the moving averages.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the MACONTROL procedure reads all the observations in a HISTORY= data set. However, if the HISTORY= data set includes the variable _PHASE_, you

can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 1936 for an example).

For an example of a HISTORY= data set, see "Creating Moving Average Charts from Subgroup Summary Data" on page 825.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC MACONTROL statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the MACONTROL procedure.

The following table lists the variables required in a TABLE= data set used with the MACHART statement:

Variable	Description
LCLE	lower control limit for Moving Average
LIMITN	nominal sample size associated with the control limits
MEAN	process mean
SPAN	number of terms in the moving average
subgroup-variable	values of the *subgroup-variable*
SUBN	subgroup sample size
SUBS	subgroup standard deviation
SUBX	subgroup mean
UCLA	upper control limit for moving average
UWMA	uniformly weighted moving average

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable of length 8.

For an example of a TABLE= data set, see "Saving Control Limit Parameters" on page 828.

Methods for Estimating the Standard Deviation

When control limits are computed from the input data, four methods are available for estimating the process standard deviation σ. Three methods (referred to as the default, MVLUE, and RMSDF) are available with subgrouped data. A fourth method is used if the data are individual measurements (see "Default Method for Individual Measurements" on page 858).

Default Method for Subgroup Samples

This method is the default for moving average charts using subgrouped data. The default estimate of σ is

$$\hat{\sigma} = \frac{s_1/c_4(n_1) + \ldots + s_N/c_4(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, s_i is the sample standard deviation of the i^{th} subgroup

$$s_i = \sqrt{\frac{1}{n_i - 1}\sum_{j=1}^{n_i}(x_{ij} - \bar{X}_i)^2}$$

and

$$c_4(n_i) = \frac{\Gamma(n_i/2)\sqrt{2/(n_i - 1)}}{\Gamma((n_i - 1)/2)}$$

Here $\Gamma(\cdot)$ denotes the gamma function, and $\bar{X}_i$ denotes the i^{th} subgroup mean. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$. If the observations are normally distributed, then the expected value of s_i is $c_4(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the *ASTM Manual on Presentation of Data and Control Chart Analysis* (1976).

MVLUE Method for Subgroup Samples

If you specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). The MVLUE is a weighted average of N unbiased estimates of σ of the form $s_i/c_4(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{h_1 s_1/c_4(n_1) + \ldots + h_N s_N/c_4(n_N)}{h_1 + \ldots + h_N}$$

where

$$h_i = \frac{[c_4(n_i)]^2}{1 - [c_4(n_i)]^2}$$

A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

RMSDF Method for Subgroup Samples

If you specify SMETHOD=RMSDF, a weighted root-mean-square estimate is computed for σ as follows:

$$\hat{\sigma} = \frac{\sqrt{(n_1 - 1)s_1^2 + \cdots + (n_N - 1)s_N^2}}{c_4(n)\sqrt{n_1 + \cdots + n_N - N}}$$

The weights are the degrees of freedom $n_i - 1$. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$.

If the unknown standard deviation σ is constant across subgroups, the root-mean-square estimate is more efficient than the minimum variance linear unbiased estimate. However, in process control applications it is generally not assumed that σ is constant, and if σ varies across subgroups, the root-mean-square estimate tends to be more inflated than the MVLUE.

Default Method for Individual Measurements

When each subgroup sample contains a single observation ($n_i \equiv 1$), the process standard deviation σ is estimated as

$$\hat{\sigma} = \sqrt{\frac{1}{2(N-1)} \sum_{i=1}^{N-1} (x_{i+1} - x_i)^2}$$

where N is the number of observations, and $x_1, x_2, \ldots, x_N$ are the individual measurements. This formula is given by Wetherill (1977), who states that the estimate of the variance is biased if the measurements are autocorrelated.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical	DATA=	*process*
Vertical	HISTORY=	subgroup mean variable
Vertical	TABLE=	_UWMA_

For example, the following sets of statements specify the label *Moving Average of Clip Gaps* for the vertical axis and the label *Day* for the horizontal axis of the moving average chart:

```
proc macontrol data=clips1;
   machart gap*day / span=4;
   label gap = 'Moving Average of Clip Gaps';
   label day = 'Day';
run;

proc macontrol history=cliphist;
   machart gap*day / span=4;
   label gapx = 'Moving Average of Clip Gaps';
   label day  = 'Day';
run;

proc macontrol table=cliptab;
   machart gap*day;
   label _uwma_   = 'Moving Average of Clip Gaps';
   label day      = 'Day';
run;
```

In this example, the label assignments are in effect only for the duration of the procedure step, and they temporarily override any permanent labels associated with the variables.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples

This section provides advanced examples of the MACHART statement.

Example 28.1. Specifying Standard Values for the Process Mean and Process Standard Deviation

See MACMA2 in the SAS/QC Sample Library

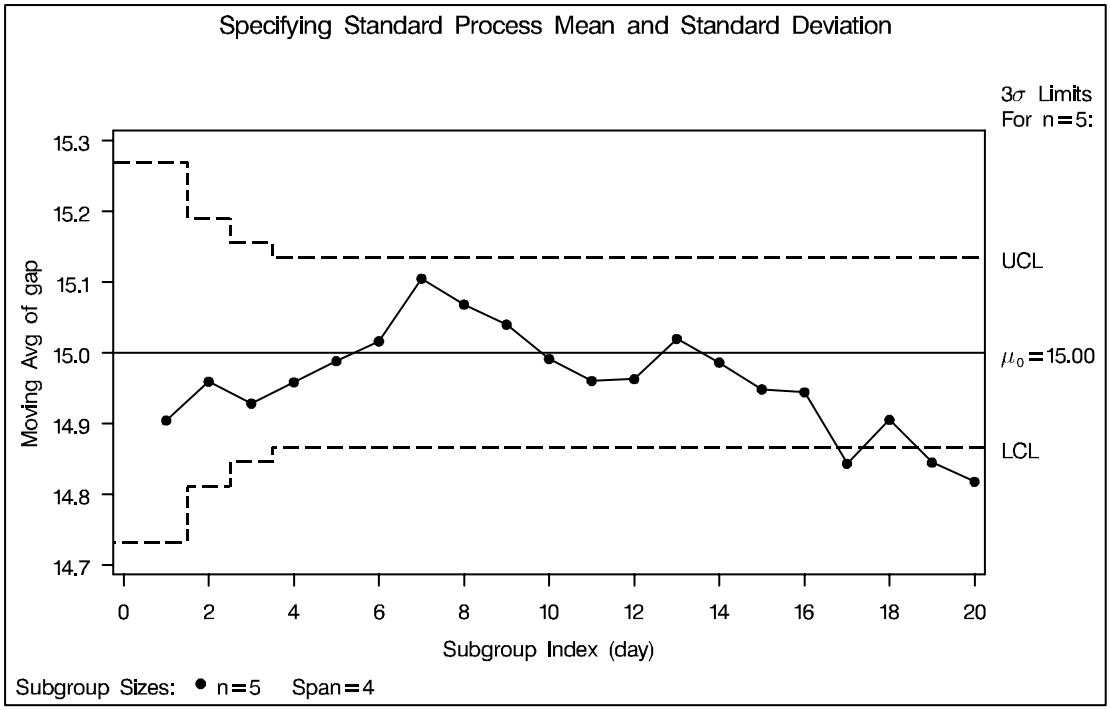
By default, the MACHART statement estimates the process mean (μ) and standard deviation (σ) from the data. This is illustrated in the "Getting Started" section of this chapter. However, there are applications in which standard values (μ_0 and σ_0) are available based, for instance, on previous experience or extensive sampling. You can specify these values with the MU0= and SIGMA0= options.

For example, suppose it is known that the metal clip manufacturing process (introduced on page 822) has a mean of 15 and standard deviation of 0.2. The following statements specify these standard values:

```
symbol h = .8;
title 'Specifying Standard Process Mean and Standard Deviation';
proc macontrol data=clips1;
   machart gap*day /
        mu0     = 15
        sigma0  = 0.2
        span    = 4
        xsymbol = mu0;
run;
```

The XSYMBOL= option specifies the label for the central line. The resulting chart is shown in Output 28.1.1.

Output 28.1.1. Specifying Standard Values with MU0= and SIGMA0=



The central line and control limits are determined using μ_0 and σ_0 (see the equations in Table 28.19 on page 846). Output 28.1.1 indicates that the process is out-of-control since the moving averages for DAY=17, DAY=19, and DAY=20 lie below the lower control limit.

You can also specify μ_0 and σ_0 with the variables _MEAN_ and _STDDEV_ in a LIMITS= data set, as illustrated by the following statements:

```
data cliplim;
   length _var_ _subgrp_ _type_ $8;
   _var_    = 'gap';
   _subgrp_ = 'day';
   _type_   = 'STANDARD';
   _limitn_ = 5;
   _mean_   = 15;
   _stddev_ = 0.2;
   _span_   = 4;
run;

proc macontrol data=clips1 limits=cliplim;
   machart gap*day / xsymbol=mu0;
run;
```

The variable _SPAN_ is required, and its value provides the number of terms in the moving average. The variables _VAR_ and _SUBGRP_ are also required, and their values must match the *process* and *subgroup-variable*, respectively, specified in the MACHART statement. The bookkeeping variable _TYPE_ is not required, but it is recommended to indicate that the variables _MEAN_ and _STDDEV_ provide standard values rather than estimated values.

The resulting chart (not shown here) is identical to the one shown in Output 28.1.1.

Example 28.2. Annotating Average Run Lengths on the Chart

See MACMA3 in the SAS/QC Sample Library

You can use Table 28.20 on page 847 and Table 28.21 on page 848 to find a moving average chart scheme with the desired average run length properties. Specifically, you can find a combination of k and w that yields a desired ARL for an in-control process ($\delta = 0$) and for a specified shift of δ.

You can also use these tables to evaluate an existing moving average chart scheme. For example, the moving average chart shown in Output 28.1.1 has a two-sided scheme with $w = 4$ and $k = 3$. Suppose you want to detect a shift of $\delta = .5$. From Table 28.21, the average run length with $w = 4$, $k = 3$, and $\delta = .5$ is 72.47. The in-control average run length ($\delta = 0$) for this scheme is 481.16.

The following statements create an inset data set that can be read to display these ARL values on the moving average chart:

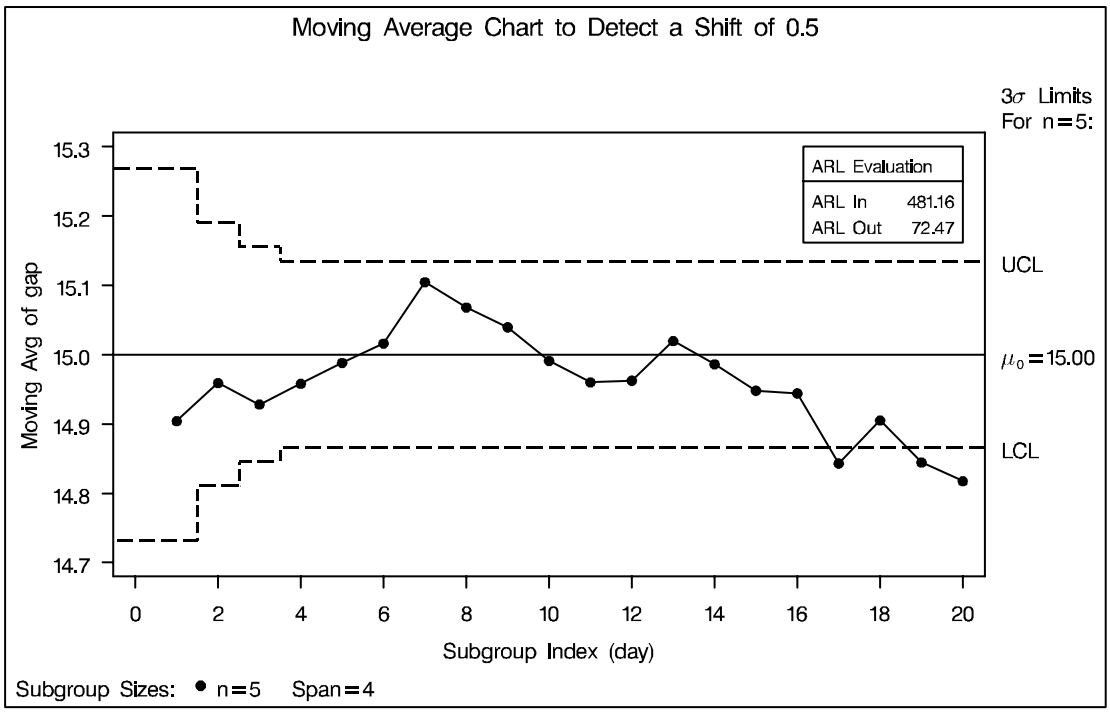
```
data arlinset;
   length _label_ $ 8;
   _label_ = 'ARL In';
   _value_ = 481.16;
   output;
   _label_ = 'ARL Out';
   _value_ = 72.47;
   output;
run;
```

The following statements create the moving average chart shown in Output 28.2.1.

```
title 'Moving Average Chart to Detect a Shift of 0.5';
symbol h = .8;
proc macontrol data=clips1;
   machart gap*day /
      mu0    = 15
      sigma0 = 0.2
      span   = 4
      xsymbol= mu0
      haxis  = axis1
      vaxis  = axis2;
   inset data = arlinset /
      header = 'ARL Evaluation'
      pos=ne height=2.5;
   axis1 offset=(2 pct, 2pct);
   axis2 offset=(2 pct, 2pct);
run;
```

The average run lengths in this example (481.16 and 72.27) are simply copied from Table 28.21. You can generalize the preceding program so that it computes the average run lengths by incorporating the simulation program on page 850.

Output 28.2.1. Displaying Average Run Lengths on Chart



For more information on annotating charts with insets, refer to Chapter 29, "INSET Statement."

Chapter 29
INSET Statement

Chapter Contents

OVERVIEW 865

GETTING STARTED 865

SYNTAX 867

Chapter 29
INSET Statement

Overview

The INSET statement allows you to enhance a moving average control chart by adding a box or table (referred to as an *inset*) of summary statistics directly to the graph. A possible application of an inset is to present moving average parameters on the chart rather than displaying them in a legend. An inset can also display arbitrary values provided in a SAS data set.

Note that the INSET statement by itself does not produce a display but must be used in conjunction with an MACHART or EWMACHART statement. Insets are not available with line printer output, so the INSET statement is not applicable when the LINEPRINTER option is specified in the PROC MACONTROL statement.

You can use options in the INSET statement to

- specify the position of the inset
- specify a header for the inset table
- specify graphical enhancements, such as background colors, text colors, text height, text font, and drop shadows

Getting Started

This section introduces the INSET statement with a basic example showing how it is used. See Chapter 52, "INSET and INSET2 Statements," for a complete description of the INSET statement.

This example is based on the same scenario as the first example in the "Getting Started" section of Chapter 27, "EWMACHART Statement." An EWMA chart is used to analyze data from the manufacture of metal clips. The following statements create a data set containing measurements to be analyzed and the EWMA chart shown in Figure 29.1.

```
data clips1;
   input day @ ;
   do i=1 to 5;
      input gap @ ;
      output;
      end;
   drop i;
   datalines;
 1  14.76  14.82  14.88  14.83  15.23
 2  14.95  14.91  15.09  14.99  15.13
 3  14.50  15.05  15.09  14.72  14.97
 4  14.91  14.87  15.46  15.01  14.99
```

```
 5  14.73  15.36  14.87  14.91  15.25
 6  15.09  15.19  15.07  15.30  14.98
 7  15.34  15.39  14.82  15.32  15.23
 8  14.80  14.94  15.15  14.69  14.93
 9  14.67  15.08  14.88  15.14  14.78
10  15.27  14.61  15.00  14.84  14.94
11  15.34  14.84  15.32  14.81  15.17
12  14.84  15.00  15.13  14.68  14.91
13  15.40  15.03  15.05  15.03  15.18
14  14.50  14.77  15.22  14.70  14.80
15  14.81  15.01  14.65  15.13  15.12
16  14.82  15.01  14.82  14.83  15.00
17  14.89  14.90  14.60  14.40  14.88
18  14.90  15.29  15.14  15.20  14.70
19  14.77  14.60  14.45  14.78  14.91
20  14.80  14.58  14.69  15.02  14.85
;

title 'EWMA Chart for Gap Measurements';
symbol v=dot;
proc macontrol data=clips1;
   ewmachart gap*day / weight   = 0.3
                       nolegend;
   inset stddev weight / cfill=blank;
run;
```

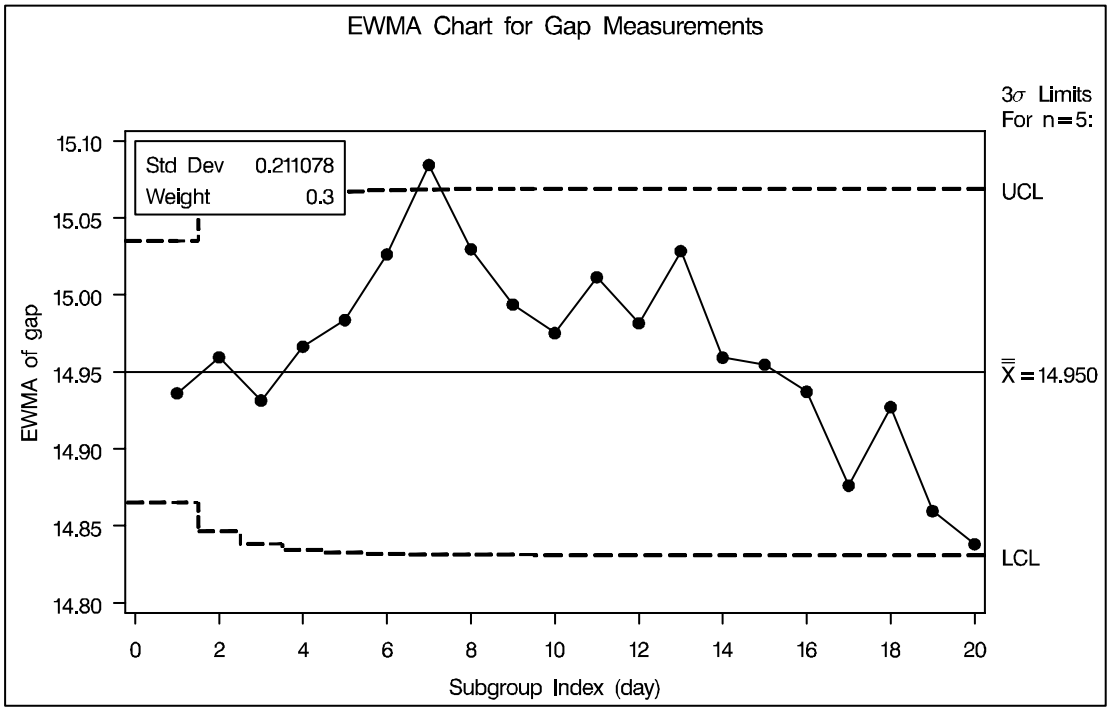


Figure 29.1. Exponentially Weighted Moving Average Chart with an Inset

Syntax

The syntax for the INSET statement is as follows:

INSET *keyword-list* < **/** *options* >**;**

You can use any number of INSET statements in the MACONTROL procedure. Each INSET statement produces a separate inset and must follow an MACHART or EWMACHART statement. The inset appears on every panel (page) produced by the last chart statement preceding it.

Keywords specify the statistics to be displayed in an inset; options control the inset's location and appearance. A complete description of the INSET statement syntax is given starting on page 1841. The INSET statement options are identical in the MACONTROL and SHEWHART procedures, but the available keywords are different. The options are listed in Table 52.5 on page 1844. The keywords available with the MACONTROL procedure are listed in Table 29.1 to Table 29.3.

Table 29.1. Summary Statistics

MEAN	estimated or specified process mean
N	nominal subgroup size
NMIN	minimum subgroup size
NMAX	maximum subgroup size
NOUT	number of subgroups outside control limits
NLOW	number of subgroups below lower control limit
NHIGH	number of subgroups above upper control limit
STDDEV	estimated or specified process standard deviation
DATA=	arbitrary values from *SAS-data-set*

Table 29.2. Parameter for Uniformly Weighted Moving Average Charts

SPAN	number of terms used to calculate moving average

Table 29.3. Parameter for Exponentially Weighted Moving Average Charts

WEIGHT	weight assigned to most recent subgroup mean in computation of the EWMA

References

American Society for Quality Control (1983), *ASQC Glossary and Tables for Statistical Quality Control*, 230 W. Wells Street, Milwaukee, Wisconsin 53203.

American Society for Testing and Materials (1976), *ASTM Manual on Presentation of Data and Control Chart Analysis*, 1916 Race Street, Philadelphia, PA 19103.

Burr, I. W. (1969), "Control Charts for Measurements with Varying Sample Sizes," *Journal of Quality Technology*, 1, 163–167.

Burr, I. W. (1976), *Statistical Quality Control Methods, Volume 16*, New York: Marcel Dekker, Inc.

Crowder, S. V. (1987a), "A Simple Method for Studying Run-length Distributions of Exponentially Weighted Moving Average Charts," *Technometrics*, 29, 401–408.

Crowder, S. V. (1987b), "Average Run Lengths of Exponentially Weighted Moving Average Charts," *Journal of Quality Technology,* 19, 161–164.

Hunter, J. S. (1986), "The Exponentially Weighted Moving Average," *Journal of Quality Technology*, 18, 203–210.

Kume, H. (1985), *Statistical Methods for Quality Improvement*, Tokyo: AOTS Chosakai, Ltd.

Montgomery, D. C. (1996), *Introduction to Statistical Quality Control, Third Edition*, New York: John Wiley & Sons, Inc.

Nelson, L. S. (1983), "The Deceptiveness of Moving Averages," *Journal of Quality Technology*, 15, 99–100.

Nelson, L. S. (1989), "Standardization of Shewhart Control Charts," *Journal of Quality Technology*, 21, 287–289.

Nelson, L. S. (1994), "Shewhart Control Charts With Unequal Subgroup Sizes," *Journal of Quality Technology*, 26, 64–67.

Roberts, S. W. (1959), "Control Chart Tests Based on Geometric Moving Averages," *Technometrics*, 1, 239–250.

Robinson, P. B. and Ho, T. Y. (1978), "Average Run Lengths of Geometric Moving Average Charts by Numerical Methods," *Technometrics*, 20, 85–93.

SAS Institute Inc. (1999), *SAS/GRAPH Software: Reference, Version 8*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1999), *SAS/ETS User's Guide: Version 8*, Cary, NC: SAS Institute Inc.

Wadsworth, H. M., Stephens, K. S., and Godfrey, A. B. (1986), *Modern Methods for Quality Control and Improvement*, New York: John Wiley & Sons, Inc.

Wetherill, G. B. (1977), *Sampling Inspection and Quality Control, Second Edition*, New York: Chapman and Hall.

Wortham, A. W. and Heinrich, G. F. (1972), "Control Charts Using Exponential Smoothing Techniques," *Annual Conference Transactions,* American Society for Quality Control, Milwaukee, Wisconsin, 451–458.

Wortham, A. W., and Ringer, L. J. (1971), "Control Via Exponential Smoothing," *The Logistics Review*, 7, 33–40.

Part 7
The OPTEX Procedure

Contents

Chapter 30. Introduction 873

Chapter 31. Details of the OPTEX Procedure 887

References 949

Chapter 30
Introduction

Chapter Contents

OVERVIEW 875
Features 875
Learning about the OPTEX Procedure 876

GETTING STARTED 877
Constructing a Nonstandard Design 877
Optimal Design Scenarios 882

Chapter 30
Introduction to the OPTEX Procedure

Overview

The OPTEX procedure searches for optimal experimental designs. You specify a set of candidate design points and a linear model, and the procedure chooses points so that the terms in the model can be estimated as efficiently as possible.

Most experimental situations call for standard designs, such as fractional factorials, orthogonal arrays, or central composites. Standard designs have assured degrees of precision and orthogonality that are important for the exploratory nature of experimentation. In some situations, however, standard designs are not available, such as when

- not all combinations of the factor levels are feasible
- the region of experimentation is irregularly shaped
- resource limitations restrict the number of experiments that can be performed
- there is a nonstandard linear or a nonlinear model

The OPTEX procedure can generate an efficient experimental design for any of these situations.

Note: Instead of using OPTEX directly, a more appropriate tool for you may be the ADX Interface. The ADX Interface, which has been completely revised in Version 7, is designed primarily for engineers and researchers who require a point-and-click solution for the entire experimental process, from building the designs through determining significant effects to optimization and reporting. In addition to offering standard designs as mentioned above, ADX makes it easy to use OPTEX to find optimal designs for non-standard factorial, response surface, and mixture experiments, with and without blocking. For more information about the ADX Interface, see Chapter 1, "Overview of ADX." (*Getting Started with the SAS 9 ADX Interface for Design of Experiments*)

Features

This section summarizes key features of the OPTEX procedure.

The OPTEX procedure offers various criteria for searching a design; these criteria are summarized in Table 30.1 on page 876 and Table 30.2 on page 876. In the formulas for these criteria, X denotes the design matrix, $\mathcal{C}$ the set of candidate points, and $\mathcal{D}$ the set of design points. The default criterion is D-optimality. You can also use the OPTEX procedure to generate G- and I-efficient designs.

The OPTEX procedure also offers a variety of search algorithms, ranging from a simple sequential search (Dykstra 1971) to the computer-intensive Fedorov algorithm

(Fedorov 1972, Cook and Nachtsheim 1980). You can customize many aspects of the search, such as the initialization method and the number of iterations.

You can use the full general linear modeling facilities of the GLM procedure to specify a model for your design, allowing for general polynomial effects as well as classification or ANOVA effects. Optionally, you can specify

- design points to be optimally augmented
- fixed covariates (for example, blocks) for the design
- prior precisions for Bayesian optimal design

The OPTEX procedure is an interactive procedure. After specifying an initial design, you can submit additional statements without reinvoking the OPTEX procedure. Once you have found a design, you can

- examine the design
- output the design to a data set
- change the model and find another design
- change the characteristics of the search and find another design

Table 30.1. Information-based Optimality Criteria

Criterion	Goal	Formula
D-optimality	Maximize determinant of the information matrix	$\max \lvert X'X \rvert$
A-optimality	Minimize sum of the variances of estimated coefficients	$\min\ \mathrm{trace}(X'X)^{-1}$

Table 30.2. Distance-based Optimality Criteria

Criterion	Goal	Formula
U-optimality	Minimize distance from design to candidates	$\min \sum_{\mathbf{x}\in\mathcal{C}} d(\mathbf{x}, \mathcal{D})$
S-optimality	Maximize distance between design points	$\min \sum_{\mathbf{y}\in\mathcal{D}} d(\mathbf{y}, \mathcal{D}-\mathbf{y})$

Learning about the OPTEX Procedure

To learn the basic syntax of the OPTEX procedure, read the introductory example in the next section, which covers a typical application of optimal designs. Other applications are illustrated in "Optimal Design Scenarios" on page 882. The summary tables in the "Summary of Functions" section on page 890 provides an overview of the syntax. The "Advanced Examples" section on page 906 illustrates construction of complex designs.

Getting Started

The examples in this section illustrate basic features of the OPTEX procedure. In addition, the examples show how a variety of SAS software tools can be used to construct candidate sets. If you are working through these examples on your own computer, note that the randomness in the OPTEX procedure's search algorithm will cause your results to be slightly different from those shown.

See "Advanced Examples" on page 906 for illustrations of complex features.

Constructing a Nonstandard Design

This example shows how you can use the OPTEX procedure to construct a design for a complicated experiment for which no standard design is available.

See OPTEXG1 in the SAS/QC Sample Library

A chemical company is designing a new reaction process. The engineers have isolated the following five factors that might affect the total yield:

Variable	Description	Range
RTEMP	Temperature of the reaction chamber	150-350 degrees
PRESS	Pressure of the reaction chamber	10-30 psi
TIME	Amount of time for the reaction	3-5 minutes
SOLV	Amount of solvent used	20-25 %
SOURCE	Source of raw materials	1, 2, 3, 4, 5

While there are only two solvent levels of interest, the reaction control factors (RTEMP, PRESS, and TIME) may be curvilinearly related to the total yield and thus require three levels in the experiment. The SOURCE factor is categorical with five levels. Additionally, some combinations of the factors are known to be problematic; simultaneously setting all three reaction control factors to their lowest feasible levels will result in worthless sludge, while setting them all to their highest levels can damage the reactor. Standard experimental designs do not apply to this situation.

Creating the Candidate Set

You can use the OPTEX procedure to generate a design for this experiment. The first step in generating an optimal design is to prepare a data set containing the candidate runs (that is, the feasible factor level combinations). In many cases, this step involves the most work. You can use a variety of SAS data manipulation tools to set up the candidate data set. In this example, the candidate runs are all possible combinations of the factor levels except those with all three control factors at their low levels and at their high levels, respectively. The PLAN procedure (refer to the *SAS/STAT User's Guide*) provides an easy way to create a full factorial data set, which can then be subsetted using the DATA step, as shown in the following statements:

```
proc plan ordered;
   factors rtemp=3 press=3 time=3 solv=2 source=5/noprint;
   output out=can
      rtemp  nvals=(150 to 350 by 100)
      press  nvals=( 10 to  30 by  10)
```

```
      time   nvals=(  3 to   5       )
      solv   nvals=( 20 to  25 by   5)
      source nvals=(  1 to   5       );
data can; set can;
   if (^((rtemp = 150) & (press = 10) & (time = 3)));
   if (^((rtemp = 350) & (press = 30) & (time = 5)));
proc print data=can;
run;
```

A partial listing of the candidate data set CAN is shown in Figure 30.1.

Obs	rtemp	press	time	solv	source
1	150	10	4	20	1
2	150	10	4	20	2
3	150	10	4	20	3
4	150	10	4	20	4
5	150	10	4	20	5
6	150	10	4	25	1
7	150	10	4	25	2
8	150	10	4	25	3
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
249	350	30	4	25	4
250	350	30	4	25	5

Figure 30.1. Candidate Set of Runs for Chemical Reaction Design

Generating the Design

The next step is to invoke the OPTEX procedure, specifying the candidate data set as the input data set. You must also provide a model for the experiment, using the MODEL statement, which uses the linear modeling syntax of the GLM procedure (refer to the *SAS/STAT User's Guide*). Since SOURCE is a classification factor, you need to specify it in a CLASS statement. To detect possible cross-product effects in the other factors, as well as the quadratic effects of the three reaction control factors, you can use a modified response surface model, as shown in the following statements:

```
proc optex data=can seed=12345;
   class source;
   model source solv|rtemp|press|time@2
         rtemp*rtemp press*press time*time;
run;
```

Note that the MODEL statement does not involve a response variable (unlike the MODEL statement in the GLM procedure). The default number of runs for a design is assumed by the OPTEX procedure to be 10 plus the number of parameters (a total of $10 + 18 = 28$ in this case.) Thus, the procedure searches for 28 runs among the candidates in CAN that allow D-optimal estimation of the effects in the model. (See "Optimality Criteria" on page 939 for a precise definition of D-optimality.) Randomness is built into the search algorithm to overcome the problem of local optima, so by default the OPTEX procedure takes 10 random "tries" to find the best design. The output, shown in Figure 30.2, lists efficiency factors for the 10 designs found. These designs are all very close in terms of their D-efficiency.

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	57.0082	32.8139	78.3162	0.8319
2	56.7660	27.3874	75.8168	0.8563
3	56.2145	28.7217	74.9937	0.8594
4	55.8960	28.7509	74.4196	0.8559
5	55.7341	29.9372	74.4554	0.8544
6	55.6224	31.4902	73.6200	0.8626
7	55.5762	28.3016	75.8959	0.8652
8	55.5080	30.3889	78.4385	0.8552
9	55.3366	28.5103	74.7014	0.8614
10	55.2176	26.8133	76.2307	0.8660

Figure 30.2. Efficiency Factors for Chemical Reaction Design

The final step is to save the best design in a data set. You can do this interactively by submitting the OUTPUT statement immediately after the preceding statements. Then use the PRINT procedure to list the design. The design is partially listed in Figure 30.3.

```
   output out=reactor;
proc print data=reactor;
run;
```

Obs	solv	rtemp	press	time	source
1	20	150	20	4	5
2	20	250	10	5	5
3	20	350	30	3	5
4	25	150	30	5	5
5	25	250	10	3	5
6	25	350	20	5	5
7	20	150	10	5	4
8	20	150	30	3	4
9	20	350	10	3	4
10	20	350	20	5	4
11	25	250	30	4	4
12	20	250	10	3	3
13	20	350	30	4	3
14	25	150	30	3	3
15	25	350	10	5	3
16	25	350	20	3	3
17	20	150	30	5	2
18	20	250	30	3	2
19	20	350	10	5	2
20	25	150	10	4	2
21	25	250	20	5	2
22	25	350	30	4	2
23	20	150	20	3	1
24	20	250	20	4	1
25	20	250	30	5	1
26	25	150	10	5	1
27	25	350	10	4	1
28	25	350	30	3	1

Figure 30.3. Optimal Design for Chemical Reaction Process Experiment

Customizing the Number of Runs

The OPTEX procedure provides options with which you can customize many aspects of the design optimization process. Suppose the budget for this experiment can only accommodate 25 runs. You can use the N= option in the GENERATE statement to request a design with this number of runs.

```
proc optex data=can seed=12345;
   class source;
   model source solv|rtemp|press|time@2
         rtemp*rtemp press*press time*time;
   generate n=25;
run;
```

Including Specific Runs

If there are factor combinations that you want to include in the final design, you can use the OPTEX procedure to *augment* those combinations optimally. For example, suppose you want to force four specific factor combinations to be in the design. If these combinations are saved in a data set, you can force them into the design by specifying the data set with the AUGMENT= option in the GENERATE statement. This technique is demonstrated in the following statements:

```
data preset;
   input solv rtemp press time source;
   datalines;
20 350 10 5 4
20 150 10 4 3
25 150 30 3 3
25 250 10 5 3
;
proc optex data=can seed=12345;
   class source;
   model source solv|rtemp|press|time@2
         rtemp*rtemp press*press time*time;
   generate n=25 augment=preset;
   output out=reactor2;
run;
```

The final design is listed in Figure 30.4 on page 881. Note that the points in the AUGMENT= data set appear as observations 7, 11, 15, and 16.

Using an Alternative Search Technique

You can also specify a variety of optimization methods with the GENERATE statement. The default method is relatively fast; while other methods may find better designs, they take longer to run and the improvement is usually only marginal. The method that generally finds the best designs is the modified Fedorov procedure described by Cook and Nachtsheim (1980). The following statements show how to request this method:

```
proc optex data=can seed=12345;
   class source;
   model source solv|rtemp|press|time@2
         rtemp*rtemp press*press time*time;
   generate n=25 method=m_fedorov;
run;
```

Obs	solv	rtemp	press	time	source
1	20	150	10	5	5
2	20	150	20	3	5
3	20	350	30	4	5
4	25	250	30	5	5
5	25	350	10	4	5
6	20	150	10	4	4
7	20	350	10	5	4
8	25	150	30	5	4
9	25	250	10	5	4
10	25	350	20	3	4
11	20	150	30	5	3
12	20	250	10	4	3
13	20	350	30	3	3
14	25	150	30	3	3
15	25	350	20	5	3
16	20	150	30	3	2
17	20	350	20	5	2
18	25	150	20	5	2
19	25	250	10	3	2
20	25	350	30	4	2
21	20	250	20	4	1
22	20	350	10	3	1
23	25	150	10	4	1
24	25	350	10	5	1
25	25	350	30	3	1

Figure 30.4. Augmented Design for Chemical Reaction Process Experiment

The efficiencies for the resulting designs are shown in Figure 30.5.

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	56.3990	29.8770	70.6062	0.9000
2	56.3897	26.3208	73.4776	0.9171
3	56.3897	26.3208	73.4776	0.9171
4	56.3758	28.1178	80.0290	0.9047
5	56.3530	32.7937	78.5367	0.8866
6	56.1983	29.5310	74.4665	0.8984
7	55.7772	28.8823	75.5397	0.9028
8	55.6893	30.8614	74.0543	0.9080
9	55.6893	30.8614	74.0543	0.9080
10	55.6893	30.8614	74.0543	0.9080

Figure 30.5. Efficiency Factors for the Modified Fedorov Search

In this case, the modified Fedorov procedure takes three to four times longer than the default method, and D-efficiency shows no improvement. On the other hand, the longer search method often does improve the design and may take only a few seconds on a reasonably fast computer.

Optimal Design Scenarios

The following examples briefly describe some additional common situations that call for optimal designs. These examples show how you can

- use a variety of SAS software tools to generate an appropriate set of candidate runs
- use the OPTEX procedure to search the candidate set for an optimal design

The emphasis here is on the programming techniques; output is omitted.

Constructing a Saturated Second-Order Design

Suppose you want a design for seven two-level factors that is as small as possible but still permits estimation of all main effects and two-factor interactions. Among standard orthogonal arrays, the smallest appropriate 2^k design has 64 runs, far more than the 29 parameters you want to estimate. To generate a D-efficient non-orthogonal design, first use the FACTEX procedure to create the full set of $2^7 = 128$ candidate runs. Then invoke the OPTEX procedure with a full second-order model, asking for a saturated design.

```
proc factex;
   factors x1-x7;
   output out=can1;

proc optex data=can1 seed=12345;
   model x1|x2|x3|x4|x5|x6|x7@2;
   generate n=saturated;
   output out=design1a;
run;
```

The default search procedure quickly finds a design with a D-efficiency of 82.3%. If search time is not an issue, you can try a more powerful search technique. For example, you can specify 500 tries with the modified Fedorov method.

```
proc optex data=can1 seed=12345;
   model x1|x2|x3|x4|x5|x6|x7@2;
   generate n=saturated
            method=m_fedorov
            iter=500;
   output out=design1b;
run;
```

This takes more than ten times longer to run, and the resulting design is only slightly more D-efficient.

Augmenting a Resolution 4 Design

In a situation similar to the previous example, suppose you have performed an experiment for seven two-level factors with a 16-run, fractional factorial design of resolution 4. You can estimate all main effects with this design, but some two-factor interactions will be confounded with each other. You now want to add enough runs to estimate all two-factor interactions as well. You can use the FACTEX procedure to create the original design as well as the candidate set.

```
proc factex;
   factors x1-x7;
   output out=can2;
run;
   model resolution=4;
   size design=min;
   output out=aug2;
run;
```

Now specify AUG2 (the data set containing the design to be augmented) with the AUGMENT= option in the GENERATE statement.

```
proc optex data=can2 seed=12345;
   model x1|x2|x3|x4|x5|x6|x7@2;
   generate n=30 augment=aug2;
   output out=design2;
run;
```

Handling Many Variables

When you have many factors, the set of all possible factor level combinations may be too large to work with as a candidate set. Suppose you want a main-effects design for 15 three-level factors. The complete set of $3^{15} > 14,000,000$ candidates is too large to use with the OPTEX procedure; in fact, it will probably be too large to store in your computer. One solution is to find a subset of the full factorial set to use as candidates. For example, an alternative candidate set is the 81-run orthogonal design of resolution 3, which can easily be constructed using the FACTEX procedure.

```
proc factex;
   factors x1-x15 / nlev=3;
   model resolution=3;
   size design=81;
   output out=can3;
proc optex data=can3 seed=12345;
   class x1-x15;
   model x1-x15;
   generate n=saturated;
   output out=design3;
run;
```

Constructing an Incomplete Block Design

An incomplete block design is a design for v qualitative treatments in b blocks of size k, where $k < v$ so that not all treatments can occur in each block. To construct an incomplete block design with the OPTEX procedure, simply create a candidate data set containing a treatment variable with t values and then use the BLOCKS statement. For example, the following statements construct a design for seven treatments in seven blocks of size three:

```
data can4;
   do treatmt = 1 to 7;
      output;
      end;
proc optex data=can4 seed=12345;
   class treatmt;
   model treatmt;
   blocks structure=(7)3;
run;
```

The resulting design is balanced in the sense that each treatment occurs the same number of times and each pair of treatments occur together in the same number of blocks. Balanced designs, when they exist, are known to be optimal, and the OPTEX procedure usually succeeds at finding them for small- to moderately-sized problems.

Constructing a Mixture-Process Design

Suppose you want to design an experiment with three *mixture factors* X1, X2, and X3 (continuous factors that represent proportions of the components of a mixture) and one *process factor* A (a classification factor with five levels). Furthermore, suppose that X1 can account for no more than 50% of the mixture. The following statements create a data set containing the vertices and generalized edge centroids of the region defined by the mixture factor constraints and then use the FACTEX procedure (see page 587) to create a candidate set that includes the process factor.

```
data xvt;
   input x1 x2 x3 @@;
datalines;
0.50 0.000 0.500
0.50 0.500 0.000
0.00 1.000 0.000
0.00 0.000 1.000
0.00 0.500 0.500
0.50 0.250 0.250
0.25 0.000 0.750
0.25 0.750 0.000
0.25 0.375 0.375
;
proc factex;
   factors a / nlev=5;
   output out=can5 pointrep=xvt;
run;
```

Analyzing mixture designs with linear models can be problematic because of the constraint that the mixture factors sum to one; however, to generate an optimal design, you can simply drop one of the mixture factors. The following statements use the preceding candidate set to find an optimal design for fitting the main effect of A and a second-order model in the mixture factors:

```
proc optex data=can5 seed=12345;
   class a;
   model a x1|x2 x1*x1 x2*x2;
run;
```

See Example 31.10 on page 926 for a more detailed example of a mixture experiment.

Chapter 31
Details of the OPTEX Procedure

Chapter Contents

SYNTAX . . . 889
Summary of Functions . . . 890

STATEMENT DESCRIPTIONS . . . 892
PROC OPTEX Statement . . . 892
BLOCKS Statement . . . 893
CLASS Statement . . . 895
EXAMINE Statement . . . 900
GENERATE Statement . . . 901
ID Statement . . . 904
MODEL Statement . . . 904
OUTPUT Statement . . . 905

ADVANCED EXAMPLES . . . 906
Example 31.1. Nonstandard Linear Model . . . 906
Example 31.2. Comparing DETMAX Algorithm to Sequential Algorithm . . . 910
Example 31.3. Using an Initial Design to Search an Optimal Design . . . 912
Example 31.4. Optimal Design Using an Augmented Best Design . . . 914
Example 31.5. Optimal Design Using a Small Candidate Set . . . 915
Example 31.6. Bayesian Optimal Design . . . 916
Example 31.7. Balanced Incomplete Block Design . . . 919
Example 31.8. Optimal Design with Fixed Covariates . . . 921
Example 31.9. Optimal Design in the Presence of Covariance . . . 924
Example 31.10. Adding Space-Filling Points to a Design . . . 926

DATA DETAILS . . . 931
Input Data Sets . . . 931
Output Data Sets . . . 933

COMPUTATIONAL DETAILS . . . 933
Specifying Effects in MODEL Statements . . . 933
Design Efficiency Measures . . . 936
Design Coding . . . 937
Optimality Criteria . . . 939
Memory and Run-Time Considerations . . . 942
Search Methods . . . 943
Optimal Blocking . . . 946

Search Strategies 946
Output 947
ODS Tables 948

Chapter 31
Details of the OPTEX Procedure

Syntax

You can specify the following statements with the OPTEX procedure. Items within the brackets <> are optional.

PROC OPTEX < *options* > **;**
CLASS *class-variables* **;**
MODEL *effects* < / *options*> **;**
BLOCKS *block-specification* <*options*> **;**
EXAMINE <*options*> **;**
GENERATE <*options*> **;**
ID *variables* **;**
OUTPUT **OUT=** *SAS-data-set* <*options*> **;**

To generate a design, you use the PROC OPTEX and MODEL statements. You can use the other statements as needed. The OPTEX procedure is interactive and allows you to use all statements (except the PROC OPTEX statement) after the first RUN statement.

Statement Ordering for Covariate Designs

You use the CLASS and MODEL statements to define a linear model for the runs in the candidate data set. You can also use these statements to define a general covariate model. In this case, list the CLASS and MODEL statements that define the model for the candidate points directly after the PROC OPTEX statement. Then list the CLASS and MODEL statements that define the covariate model after the BLOCKS DESIGN= specification. Thus, in this case, the ordering for these statements should be

1. PROC OPTEX statement
2. CLASS and MODEL statements for the candidate points
3. BLOCKS DESIGN= statement
4. CLASS and MODEL statements for the covariates

Note also that a CLASS statement naming classification variables must precede the MODEL statement that uses those variables.

Summary of Functions

Table 31.1, Table 31.2, and Table 31.3 classify the OPTEX statements and options by function.

Table 31.1. Summary of Options for Specifying the Design

Function	Statement	Option
Design Characteristics		
Number of design points	GENERATE	N=*number*
Saturated design	GENERATE	N=SATURATED
Augmented design	GENERATE	AUGMENT=*SAS-data-set*
Bayesian optimal design	MODEL	/ PRIOR=$p_1, p_2, \ldots$
Optimality Criteria		
Minimize trace of $(X'X)^{-1}$	GENERATE	CRITERION=A
Maximize $\|X'X\|$	GENERATE	CRITERION=D
Minimize mean minimum distance to design	GENERATE	CRITERION=U
Maximize mean distance between nearest design points	GENERATE	CRITERION=S
Model Specification		
Specify independent effects	MODEL	*effects*
Exclude intercept term	MODEL	*effects* NOINT
Specify class variables	CLASS	*variables*
Specify class variable parameterization	CLASS	/ PARAM=*parameterization*
Display class variable parameterization	PROC OPTEX	CLASSPARAM
Static coding	PROC OPTEX	CODING=STATIC
Orthogonal coding	PROC OPTEX	CODING=ORTH
Orthogonal coding with respect to candidates only	PROC OPTEX	CODING=ORTHCAN
Suppress coding of effects	PROC OPTEX	NOCODE
Block Specification		
Specify general covariance matrix for runs	BLOCKS	COVAR=*SAS-data-set* <*options*> VAR=*variables*
Specify general covariate model	BLOCKS	DESIGN=*SAS-data-set* <*options*>
Specify b blocks of size k	BLOCKS	STRUCTURE=$(b)k$ <*options*>
Options for block specifications		
Repeat the search n times		ITER=n
Retain best m searches		KEEP=m
Select initial design at random		INIT=RANDOM
Select initial design in order		INIT=CHAIN
Initial Design Characteristics		
Random and sequential methods	GENERATE	INITDESIGN=PARTIAL< (m) >
Random initial design	GENERATE	INITDESIGN=RANDOM

Function	Statement	Option
Sequential initial design	GENERATE	INITDESIGN=SEQUENTIAL
Specify initial design	GENERATE	INITDESIGN=*SAS-data-set*

Table 31.2. Summary of Options for Searching for the Design

Function	Statement	Option
Design Search Specification		
Retain best n searches	GENERATE	KEEP=n
Search n times	GENERATE	ITER=n
Specify candidate points	PROC OPTEX	DATA=*SAS-data-set*
Specify random seed	PROC OPTEX	SEED=*number*
Specify effective zero	PROC OPTEX	EPSILON=ϵ
Design Search Methods		
DETMAX algorithm with maximum excursion *level*	GENERATE	METHOD=DETMAX<*(level)*>
Exchange algorithm	GENERATE	METHOD=EXCHANGE
k-Exchange algorithm	GENERATE	METHOD=EXCHANGE$<(k)>$
Sequential algorithm	GENERATE	METHOD=SEQUENTIAL
Fedorov algorithm	GENERATE	METHOD=FEDOROV
Modified Fedorov algorithm	GENERATE	METHOD=M_FEDOROV

Table 31.3. Summary of Options for Examining and Saving the Design

Function	Statement	Option
Save the Design		
Best design	OUTPUT OUT=*SAS-data-set*	
Specific design	OUTPUT OUT=*SAS-data-set*	NUMBER=*design-number*
Block variable name	OUTPUT OUT=*SAS-data-set*	BLOCK=*variable-name*
Specify transfer variables	ID	*variables*
List the Design		
Design characteristics	EXAMINE	
Design points	EXAMINE	DESIGN
Information matrix $X'X$	EXAMINE	INFORMATION
Specific optimal design	EXAMINE	NUMBER=*design-number*
Variance matrix $(X'X)^{-1}$	EXAMINE	VARIANCE
Suppress all output	PROC OPTEX	NOPRINT

Statement Descriptions

This section provides detailed syntax information for the OPTEX procedure statements, beginning with the PROC OPTEX statement. The remaining statements are presented in alphabetical order.

PROC OPTEX Statement

PROC OPTEX <*options*> **;**

You use the PROC OPTEX statement to invoke the procedure. The following *options* can be used:

CLASSPARAM
specifies that a table should be displayed summarizing the parameterization of classification variables in the model for the design.

CODING=NONE
CODING=STATIC
CODING=ORTH
CODING=ORTHCAN
specifies which type of coding to use for modeling effects in the design. Coding equalizes all model effects as far as the optimization is concerned. The default is CODING=STATIC, which specifies that the values of all effects are to be coded to have maximum and minimum values of $+1$ and -1, respectively. The options CODING=ORTH and CODING=ORTHCAN specify orthogonal coding with respect to the input points. The option CODING=NONE suppresses coding of effects; it is equivalent to the NOCODE option. For more details on coding, see "Design Coding" on page 937.

Note that while CODING=STATIC is the default, CODING=ORTH will usually give more appropriate efficiency values, especially if all possible combinations of factor levels occur in the candidate data set.

DATA=*SAS-data-set*
specifies the input SAS data set that contains the candidate points for the design. By default, the OPTEX procedure uses the most recently created SAS data set. For details, see "DATA= Data Set" on page 931.

EPSILON=ϵ
specifies the smallest value ϵ that is considered to be nonzero for determining when the search is no longer yielding an improved design and when the information matrix for the design is singular. By default, $\epsilon = 0.00001$.

NAMELEN=*n*
specifies the length of effect names in tables and output data sets to be n characters long, where n is a value between 20 and 200 characters. The default length is 20 characters.

NOCODE
suppresses the coding of effects in the model for the design. This option is equivalent to CODING=NONE.

NOPRINT
suppresses all output. This is useful when you only want the final design to be saved in a data set.

SEED=*s*
specifies an integer used to start the pseudo-random number generator for initialization (see "Search Methods" on page 943). If you don't specify a seed, or specify a value less than or equal to zero, the seed is by default generated from reading the time of day from the computer's clock.

STATUS=*status-level*
specifies that the status of the search be checked at the given level, where *status-level* is an integer between 1 and 4, inclusive. If you specify a *status-level* then a table of the status at each check point is displayed. You can use this table to track the progress of long searches. The allowable *status-levels* are listed in the following table:

Status-level	Checks status after each:
1	design search; the number of searches specified by the NITER= option
2	search loop
3	internal search loop
4	extra internal search loop for METHOD=M_FEDOROV

Each search method loops to produce successively better designs; these are the search loops for STATUS=2. STATUS=3 and STATUS=4 refer to deeper loops within the search methods. You will only need to specify STATUS=3 or STATUS=4 very rarely, since unless simply evaluating a potential switch is very expensive (as it can occasionally be with the space-filling criteria). Evaluating and displaying the status at this level will make the search much, much slower.

BLOCKS Statement

BLOCKS *block-specification* < *options* > **;**

You use the BLOCKS statement to find a D-optimal design in the presence of fixed covariates (for example, blocks) or covariance. The technique is an extension of the optimal blocking technique of Cook and Nachtsheim (1989); see "Optimal Blocking" on page 946.

For the purposes of optimal blocking, the model for the original candidate points is referred to as the *treatment model*; the candidate points for the part of the design matrix corresponding to the treatment model form the *treatment set*. If the GENERATE statement is not specified, then the full candidate set is used as the treatment set; otherwise, an optimal design for the treatment model ignoring the blocks is first generated, and the result is used as the treatment set for optimal blocking.

The following are three mutually exclusive *block-specifications* that you can provide:

COVAR=*SAS-data-set* **VAR=(** *variables* **)**
specifies a data set to use in providing a general covariance matrix for the runs. The argument to VAR= names the variables in this data set that contain the columns of the covariance matrix for the runs. For an example, see Example 31.9 on page 924.

DESIGN=*SAS-data-set*
specifies a data set to use in providing a general covariate model. In addition to this data set, you must specify a covariate model with the CLASS and MODEL statements. Covariate models are specified in the same way as the treatment model; CLASS and MODEL statements that come after a BLOCKS statement involving the DESIGN= specification are interpreted as applying to the covariate model. For an example, see Example 31.8 on page 921.

STRUCTURE=(b**)** k
specifies a block design with b blocks of size k. For an example, see Example 31.7 on page 919.

The following *options* can also be used:

INIT=RANDOM
specifies the initialization method for constructing the starting design. The option INIT=RANDOM specifies that the starting design is to be constructed by selecting candidates at random without replacement. The option INIT=CHAIN selects candidate points in the order in which they occur in the original data set.

ITER=n
specifies the number of times to repeat the search from different initial designs. Because local optima are common in difficult search problems, it is often a good idea to make several tries for the optimal design with a random or partially random method of initialization (see the preceding INIT= option). By default, $n = 10$. You can specify ITER=0 to evaluate the initial design itself.

KEEP=m
specifies that only the best m designs are to be retained. The value m must be less than or equal to the value n of the ITER= option; by default $m = n$, so that all iterations are kept. This option is useful when you want to make many searches to overcome the problem of local optima but you are only interested in the results of the best m designs.

NOEXCHANGE
suppresses the part of the optimal blocking algorithm that exchanges treatment design points for candidate treatment points. When this option is specified, only interchanges between design points are performed. Use this option when you do not want to change which treatment points are included in the design and you only want to find their optimal ordering.

CLASS Statement

CLASS variable <(v-options)> <variable <(v-options)>... >
< / v-options > ;

You use the CLASS statement to identify classification variables, which are factors that separate the observations into groups. For example, a completely randomized design has a single *class-variable* that identifies the groups of observations. A randomized complete block design has two *class-variables*; one identifies the blocks and one identifies the treatments.

You can specify various *v-options* for each variable by enclosing them in parentheses after the variable name. You can also specify global *v-options* for the CLASS statement by placing them after a slash (/). Global *v-options* are applied to all the variables specified in the CLASS statement. However, individual CLASS variable *v-options* override the global *v-options*.

Class-variables can be either numeric or character. The OPTEX procedure uses the formatted values of *class-variables* in forming model effects. Any variable in the model that is not listed in the CLASS statement is assumed to be continuous. Continuous variables must be numeric.

Note: If you specify a data set containing fixed covariate effects with a DESIGN= data set in the BLOCKS statement, then a CLASS or MODEL statement that follows the BLOCKS statement refers to the model for the fixed covariates. A CLASS or MODEL statement that defines the model for the candidate points (treatment model) should be specified *before* the BLOCKS statement.

DESCENDING
DESC
reverses the sorting order of the classification variable.

ORDER=DATA | FORMATTED | FREQ | INTERNAL
specifies the sorting order for the levels of classification variables. This ordering determines which parameters in the model correspond to each level in the data, so the ORDER= option may be useful when you use the CONTRAST statement. When ORDER=FORMATTED (the default) for numeric variables for which you have supplied no explicit format (that is, for which there is no corresponding FORMAT statement in the current PROC OPTEX run or in the DATA step that created the data set), the levels are ordered by their internal (numeric) value. Note that this represents a change from previous releases for how class levels are ordered. In releases previous to Version 8, numeric class levels with no explicit format were ordered by their BEST12. formatted values, and in order to revert to the previous ordering you can specify this format explicitly for the affected classification variables. The change was implemented because the former default behavior for ORDER=FORMATTED often resulted in levels not being ordered numerically. The following table shows how PROC OPTEX interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=FORMATTED. For FORMATTED and INTERNAL, the sort order is machine dependent. For more information on sorting order, see the chapter on the SORT procedure in the *SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

PARAM=*keyword*

specifies the parameterization method for the classification variable or variables. Design matrix columns are created from CLASS variables according to the following coding schemes. The default is PARAM=ORTHEFFECT. Note that this represents a change from previous releases for how classification variables are parameterized. In releases previous to Version 9, the default was PARAM=EFFECT, and in order to revert to the previous parameterization you can specify PARAM=EFFECT explicitly for the affected classification variables. The change was implemented because an orthogonal parameterization leads to absolute D- and A-efficiency values that more realistically reflect the true efficiency of the design. If PARAM=ORTHPOLY or PARAM=POLY, and the CLASS levels are numeric, then the ORDER= option in the CLASS statement is ignored, and the internal, unformatted values are used.

EFFECT	specifies effect coding
POLYNOMIAL | POLY	specifies polynomial coding
REFERENCE | REF	specifies reference cell coding
ORDINAL | ORD	specifies ordinal, or “thermometer” coding
ORTHEFFECT	specifies orthogonal effect coding
ORTHPOLY	specifies orthogonal polynomial coding
ORTHREF	specifies orthogonal reference cell coding
ORTHORDINAL	specifies orthogonal ordinal coding

All of these parameterizations are full rank. The orthogonal versions perform a scaled, intercept-augmented Gram-Schmidt orthogonalization on the columns of the corresponding non-orthogonal parameterizations. For the EFFECT and REFERENCE parameterizations, the REF= option in the CLASS statement determines the reference level.

Consider a model with one CLASS variable A with four levels, 1, 2, 5, and 7. Details of the possible choices for the PARAM= option follow.

EFFECT

Three columns are created to indicate group membership of the nonreference levels. For the reference level, all three dummy variables have a value of -1. For instance, if the reference level is 7 (REF=7), the design matrix columns for A are as follows.

Effect Coding			
A	**Design Matrix**		
1	1	0	0
2	0	1	0
5	0	0	1
7	-1	-1	-1

Parameter estimates of CLASS main effects using the effect coding scheme estimate the difference in the effect of each nonreference level compared to the average effect over all 4 levels.

POLYNOMIAL

POLY

Three columns are created. The first represents the linear term (x), the second represents the quadratic term (x^2), and the third represents the cubic term (x^3), where x is the level value. If the CLASS levels are not numeric, they are translated into 1, 2, 3, ... according to their sorting order. The design matrix columns for A are as follows.

Polynomial Coding			
A	**Design Matrix**		
1	1	1	1
2	2	4	8
5	5	25	125
7	7	49	343

REFERENCE

REF

Three columns are created to indicate group membership of the nonreference levels. For the reference level, all three dummy variables have a value of 0. For instance, if the reference level is 7 (REF=7), the design matrix columns for A are as follows.

Reference Coding			
A	**Design Matrix**		
1	1	0	0
2	0	1	0
5	0	0	1
7	0	0	0

Parameter estimates of CLASS main effects using the reference coding scheme estimate the difference in the effect of each nonreference level compared to the effect of the reference level.

ORDINAL

ORD

Three columns are created to indicate group membership in successive collections of levels after the first. For instance, the design matrix columns for A are as follows.

Ordinal Coding			
A	**Design Matrix**		
1	0	0	0
2	1	0	0
5	1	1	0
7	1	1	1

Parameter estimates of CLASS main effects using the ordinal coding scheme estimate the difference in the average effect of each successive collection of levels compared to the effect of the first level.

ORTHEFFECT

The columns are obtained by applying the Gram-Schmidt orthogonalization to the mean-centered columns for PARAM=EFFECT, and then scaling so that the sum of squares for each column equals the number of levels. The design matrix columns for A are as follows.

Orthogonal Effects Coding			
A	**Design Matrix**		
1	1.414	−0.816	−0.577
2	0	1.633	−0.577
5	0	0	1.732
7	−1.414	−0.816	−0.577

ORTHPOLY

The columns are obtained by applying the Gram-Schmidt orthogonalization to the mean-centered columns for PARAM=POLY, and then scaling so that the sum of squares for each column equals the number of levels. The design matrix columns for A are as follows.

Orthogonal Polynomial Coding			
A	**Design Matrix**		
1	−1.153	0.907	−0.921
2	−0.734	−0.540	1.473
5	0.524	−1.370	−0.921
7	1.363	1.004	0.368

ORTHREF The columns are obtained by applying the Gram-Schmidt orthogonalization to the mean-centered columns for PARAM=REFERENCE, and then scaling so that the sum of squares for each column equals the number of levels. The design matrix columns for A are as follows.

Orthogonal Reference Coding			
A	**Design Matrix**		
1	1.732	0	0
2	−0.577	1.633	0
5	−0.577	−0.816	1.414
7	−0.577	−0.816	−1.414

ORTHORDINAL The columns are obtained by applying the Gram-Schmidt orthogonalization to the mean-centered columns for PARAM=REFERENCE, and then scaling so that the sum of squares for each column equals the number of levels. The design matrix columns for A are as follows.

Orthogonal Ordinal Coding			
A	**Design Matrix**		
1	−1.732	0	0
2	0.577	−1.633	0
5	0.577	0.816	−1.414
7	0.577	0.816	1.414

REF=*'level'* **|** *keyword*

specifies the reference level for PARAM=EFFECT or PARAM=REFERENCE. For an individual (but not a global) variable REF= *option*, you can specify the *level* of the variable to use as the reference level. For a global or individual variable REF= *option*, you can use one of the following *keywords*. The default is REF=LAST.

FIRST designates the first ordered level as reference

LAST designates the last ordered level as reference

TRUNCATE

specifies that class levels should be determined using only up to the first 16 characters of the formatted values of CLASS variables. When formatted values are longer than 16 characters, you can use this option in order to revert to the levels as determined in releases previous to Version 9.

EXAMINE Statement

EXAMINE <*options*> **;**

You use the EXAMINE statement to display the characteristics of a selected design. By default, the EXAMINE statement lists certain measures of design efficiency for the best design. (See the "Output" section on page 947.) The following *options* can be used to modify the output:

DESIGN
lists the actual points in the selected design.

INFORMATION
INFO
I
lists the information matrix $X'X$ for the selected design.

NUMBER=*design-number*
selects a design to examine by specifying its *design-number*. Designs are ordered by the value of the efficiency criterion that is being optimized. Thus, a *design-number* of 1 corresponds to the best design found, a *design-number* of 2 corresponds to the second best design, and so on. The default *design-number* is 1. To modify the number of designs created, see the ITER= option on page 903.

VARIANCE
VAR
V
lists the variance matrix $(X'X)^{-1}$ for the parameter estimates for the selected design.

For details on design efficiencies, see "Design Efficiency Measures" on page 936.

If you use the OPTEX procedure interactively, you must enter the options for every EXAMINE statement. For example, the following statements list default information and the design points for the best design but only default information for the second-best design:

```
examine number=1 design;
examine number=2;
```

The following statements list default information and design points for both the best and second-best designs:

```
examine number=1 design;
examine number=2 design;
```

GENERATE Statement

GENERATE <*options*> **;**

You use the GENERATE statement to customize the search for a design. By default, the OPTEX procedure searches for a design as follows:

- using the exchange algorithm (METHOD=EXCHANGE)
- using D-optimality as the optimality criterion (CRITERION=D)
- using a completely random initial design to start the search (INITDESIGN=RANDOM)
- selecting candidate points only from the DATA= data set (modified by using AUGMENT= or INITDESIGN= data sets)
- performing 10 iterations in the search (ITER=10)
- finding a design with $10 + p$ points, where p is the number of parameters in the model (modified by using the N= or INITDESIGN= option)

The following *options* can be used to modify these defaults:

AUGMENT=*SAS-data-set*
specifies a data set that contains a design to be augmented, in other words, a set of points that must be contained in the design generated. When creating designs, the OPTEX procedure adds points from the DATA= data set (or the last data set created, if DATA= is not specified) to points from the AUGMENT= data set. The number of points in the design to be augmented must be less than the number of points specified with the N= option. For details, see "AUGMENT= Data Set" on page 932.

CRITERION=*crit*
specifies the optimality criterion used in the search. You can specify any one of the following:

CRITERION=D
specifies D-optimality; the optimal design maximizes the determinant $|X'X|$ of the information matrix for the design. This is the default criterion.

CRITERION=A
specifies A-optimality; the optimal design minimizes the sum of the variances of the estimated parameters for the model, which is the same as minimizing the trace of $(X'X)^{-1}$.

CRITERION=U
specifies U-optimality; the optimal design minimizes the sum of the minimum distances from each candidate point to the design. That is, if $\mathcal{C}$ is the set of candidate points, $\mathcal{D}$ is the set of design points, and $d(\mathbf{x}, \mathcal{D})$ is the minimum distance from $\mathbf{x}$ to any point in $\mathcal{D}$, then a U-optimal design minimizes

$$\sum_{\mathbf{x} \in \mathcal{C}} d(\mathbf{x}, \mathcal{D})$$

This measures how well the design "covers" the candidate set; thus, a U-optimal design is also called a *uniform coverage design*.

CRITERION=S

specifies S-optimality; the optimal design maximizes the harmonic mean of the minimum distance from each design point to any other design point. Mathematically, an S-optimal design maximizes

$$\frac{N_D}{\sum_{\mathbf{y}\in\mathcal{D}} 1/d(\mathbf{y}, \mathcal{D}-\mathbf{y})}$$

where $\mathcal{D}$ is the set of design points, and N_D is the number of points in $\mathcal{D}$. This measures how spread out the design points are; thus, an S-optimal design is also called a *maximum spread design*.

For more information on the different criteria, see "Optimality Criteria" on page 939.

INITDESIGN=*initialization-method*

specifies a method of obtaining an initial design for the search procedure. Valid values of *initialization-method* are as follows:

SEQUENTIAL

specifies an initial design chosen by a sequential search. The design given by INITDESIGN=SEQUENTIAL is the same as the design given by METHOD=SEQUENTIAL. You can use the INITDESIGN=SEQUENTIAL option with other values of the METHOD= option to specify a sequential design as the initial design for various search methods. For details, see "Search Methods" on page 943.

RANDOM

specifies a completely random initial design. The initial design generated consists of a random selection of observations from the DATA= data set.

PARTIAL<**(***m* **)**>

specifies an initial design using a mixture of RANDOM and SEQUENTIAL methods. A small number n_r of points for the initial design are chosen at random from the candidates, and the rest of the design points are chosen by a sequential search. (For a definition of the sequential search, see "Search Methods" on page 943.)

By default, n_r is randomly chosen between 0 and half the number of parameters in the linear model. You can specify the optional integer m to modify the selection of n_r. If $m > 0$, then n_r is randomly chosen between 0 and m for each try. If $m < 0$, then $n_r = |m|$ for each try. The maximum value for $|m|$ is the number of points in the design. Refer to Galil and Kiefer (1980) for notes on choosing n_r.

SAS-data-set

specifies a data set that holds the initial design. Use this *initialization-method* when you have a specific design that you want to improve or when you want to evaluate an existing design. For details, see "INITDESIGN= Data Set" on page 932.

The default initialization method depends on the search procedure as shown in Table 31.4.

Table 31.4. Default Initialization Methods

Search Procedure (METHOD= option)	**Default Initialization Method** (INITDESIGN= option)
DETMAX	PARTIAL
EXCHANGE	RANDOM
FEDOROV	RANDOM
M_FEDOROV	PARTIAL
SEQUENTIAL	none

If you specify INITDESIGN=*SAS-data-set* and METHOD=SEQUENTIAL, no search is performed; the INITDESIGN= data set is taken as the final design. By specifying these options, you can use the procedure to evaluate an existing design.

ITER=n

specifies the number n of searches to make. Because local optima are common in difficult search problems, it is often a good idea to make several tries for the optimal design with a random or partially random method of initialization (see the preceding INITDESIGN= option). By default, $n = 10$.

The n designs found are sorted by their respective efficiencies according to the current optimality criterion (see the CRITERION=option on page 901.) The most efficient design is assigned a *design-number* of 1, the second most efficient design is assigned a *design-number* of 2, and so on. You can then use the *design-number* in the EXAMINE and OUTPUT statements to display the characteristics of a design or to save a design in a data set.

KEEP=m

specifies that only the best m designs are to be retained. The value m must be less than or equal to the value n of the ITER= option; by default $m = n$, so that all iterations are kept. This option is useful when you want to make many searches to overcome the problem of local optima but are interested only in the results of the best m designs.

METHOD=DETMAX<*(level)*>
METHOD=EXCHANGE< (k) >
METHOD=FEDOROV
METHOD=M_FEDOROV
METHOD=SEQUENTIAL

specifies the procedure used to search for the optimal design. The default is METHOD=EXCHANGE.

With METHOD=DETMAX, the optional *level* gives the maximum excursion level for the search, where *level* is an integer greater than or equal to 1. Enclose the value

of *level* in parentheses immediately following the word DETMAX. The default value for *level* is 4. In general, larger values of *level* result in longer search times.

When METHOD=EXCHANGE, the optional k specifies the k-exchange search method of Johnson and Nachtsheim (1983), which generalizes the modified Fedorov search algorithm of Cook and Nachtsheim (1980). Enclose the value of k in parentheses immediately following the word EXCHANGE.

From fastest to slowest, the methods are

$$\text{SEQUENTIAL} \rightarrow \text{EXCHANGE} \rightarrow \text{DETMAX} \rightarrow \text{M\_FEDOROV} \rightarrow \text{FEDOROV}$$

In general, slower methods result in more efficient designs. While the default method EXCHANGE always works relatively quickly, you may want to specify a more reliable method, such as M_FEDOROV, with fast computers or small- to moderately-sized problems.

See "Search Methods" on page 943 for details on the algorithms.

N=*n*
N=SATURATED
specifies the number of points in the final design. The default design size is $10 + p$, where p is the number of parameters in the model. If you use the INITDESIGN= option, the default number is the number of points in the initial design. Specify N=n to search for a design with n points. Specify N=SATURATED to search for a design with the same number of points as there are parameters in the model. A saturated design has no degrees of freedom to estimate error and should be used with caution.

ID Statement

ID *variables* **;**

You use the ID statement to name the *variables* in the DATA= data set that are not involved in the model but are to be transferred from the input data set to the output data set.

Variables listed in the ID statement must be contained in the DATA= data set. They can also be contained in other input data sets. If an ID variable is also contained in an AUGMENT= or INITDESIGN= data set and an observation from that data set is used in the final design, the values of the ID variables for that observation are transferred to the OUT= data set. For details, see "Input Data Sets" on page 931.

MODEL Statement

MODEL *effects* < **/** *options* > **;**

You use the MODEL statement to specify the independent effects used to model data that are to be collected with the design that is being constructed. The *effects* can be

- simple continuous regressor effects
- polynomial continuous effects
- main effects of classification variables
- interactions of classification variables
- continuous-by-class effects

The variables used to form *effects* in the MODEL statement must be present in all input data sets. For details on input data sets, see "Input Data Sets" on page 931. For details on the specification of different types of effects and on how the design matrix is defined with respect to the effects, see "Specifying Effects in MODEL Statements" on page 933.

If you specify a data set containing fixed covariate effects with a DESIGN= data set in the BLOCKS statement, then a CLASS or MODEL statement that *follows* the BLOCKS statement refers to the model for the fixed covariates. A CLASS or MODEL statement that defines the model for the candidate points (treatment model) should occur *before* the BLOCKS statement.

The following options can be used in the MODEL statement:

NOINT
excludes the intercept parameter from the model. By default, the OPTEX procedure includes the intercept parameter in the model.

PRIOR=*num-list*
specifies prior precision values corresponding to groups of effects in the model. Groups of effects in the MODEL statement with the same prior precision must be separated by commas. Then use the PRIOR= option, listing as many prior precision values as there are groups of effects. See Example 31.6 on page 916 for an example.

When you specify prior precision values, the information matrix for estimating the linear parameters is $X'X + P$, where X is the design matrix and P is a diagonal matrix with the prior precision values that you specify on the diagonal. Thus, in terms of a prior distribution, the inverses of the prior precision values can be interpreted as prior variances for the linear parameters corresponding to each effect. As an alternative interpretation, note that with orthogonal coding the value of the prior for an effect says roughly how many prior "observations' worth" of information you have for that effect. See "Design Coding" on page 937 for details on orthogonal coding.

OUTPUT Statement

OUTPUT OUT= *SAS-data-set* < *options* > **;**

You use the OUTPUT statement to save a design in an output data set. By default, the saved design is the best design found. You specify the data set name as follows:

OUT=*SAS-data-set*
gives a name for the output data set. The OUT= data set is required in the OUTPUT statement.

The following *options* can be used:

BLOCKNAME=*variable-name*
specifies the name to be given to the blocking variable in the output data set. The default name is BLOCK. You can use this *option* in conjunction with a STRUCTURE= option in the BLOCKS statement. See Example 31.7 on page 919 for an example.

NUMBER=*design-number*
selects a design to output by specifying its *design-number*. Designs are ordered by the value of the efficiency criterion that is being optimized. Thus, a *design-number* of 1 corresponds to the best design found, a *design-number* of 2 corresponds to the second best design, and so on. The default *design-number* is 1. To modify the number of designs created, see the ITER= option on page 903.

Alternatively, you can specify one of the following:

NUMBER=DBEST
selects the design that has the highest D-efficiency value.

NUMBER=ABEST
selects the design that has the highest A-efficiency value.

NUMBER=GBEST
selects the design that has the highest G-efficiency value.

NUMBER=VBEST
selects the design that has the minimum average standard error for prediction.

These options can be used to find designs that are efficient for more than one criterion For example, you can use the default CRITERION=D option in the GENERATE statement with the NUMBER=GBEST option in the OUTPUT statement to find the D-optimal design that has maximal G-efficiency. In fact, this is the best way to use the OPTEX procedure to find G-efficient designs; see "G- and I-optimality" on page 941 for more details.

Advanced Examples

Example 31.1. Nonstandard Linear Model

See OPTEX3 in the SAS/QC Sample Library

The following example is based on an example in Mitchell (1974a). An animal scientist wants to compare wildlife densities in four different habitats over a year. However, due to the cost of experimentation, only 12 observations can be made. The following model is postulated for the density $y_j(t)$ in habitat j during month t:

$$y_j(t) \quad = \quad \mu_j + \beta t + \sum_{i=1}^{4} a_i \cos(i\pi t/4) + \sum_{i=1}^{3} b_i \sin(i\pi t/4).$$

This model includes the habitat as a classification variable, the effect of time with an overall linear drift term βt, and cyclic behavior in the form of a Fourier series. There is no intercept term in the model.

The OPTEX procedure is used since there are no standard designs that cover this situation. The candidate set is the full factorial arrangement of four habitats by 12 months, which can be generated with a DATA step, as follows:

```
data a;
   drop theta pi;
   array c{4} c1-c4;
   array s{3} s1-s3;
   pi = arcos(-1);
   do habitat=1 to 4;
      do month=1 to 12;
         theta = pi * month / 4;
         do i=1 to 4; c{i} = cos(i*theta); end;
         do i=1 to 3; s{i} = sin(i*theta); end;
         output;
      end;
   end;
run;
```

Data set A contains the 48 candidate points and includes the cosine variables (C1, C2, C3, and C4) and sine variables (S1, S2, S3, S4). The following statements produce Output 31.1.1:

```
proc optex seed=193030034 data=a;
   class    habitat;
   model    habitat month c1-c4 s1-s3 / noint;
   generate n=12;
run;
```

Output 31.1.1. Sampling Wildlife Habitats Over Time

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	31.6103	19.7379	57.7350	1.3229
2	31.6103	19.7379	57.7350	1.3229
3	31.6103	19.7379	57.7350	1.3229
4	31.6103	19.3793	57.7350	1.3229
5	31.6103	19.2916	57.7350	1.3229
6	31.6103	19.0335	57.7350	1.3229
7	30.1304	14.4796	44.7214	1.4907
8	30.1304	14.2433	44.7214	1.5092
9	30.1304	13.1687	44.7214	1.5456
10	28.1616	9.8842	40.8248	1.7559

The best determinant (D-efficiency) was found in 6 out of the 10 tries. Thus, you can be confident that this is the best achievable determinant. Only the A-efficiency distinguishes among the designs listed in Output 31.1.1. The best design has an A-efficiency of 19.74%, whereas another design has the same D-efficiency but a slightly smaller A-efficiency of 19.03%, or about 96% relative A-efficiency. To explore the differences, you can save the designs in data sets and print them. Since the OPTEX procedure is interactive, you need to submit only the following statements (immediately after the preceding statements) to produce Output 31.1.2 and Output 31.1.3:

```
   output out=d1 number=1;
run;
   output out=d6 number=6;
run;

proc sort data=d1;
   by  month habitat;
proc print data=d1;
   var month habitat;
run;

proc sort data=d6;
   by  month habitat;
proc print data=d6;
   var month habitat;
run;
```

Output 31.1.2. The Best Design

Obs	month	habitat
1	1	3
2	2	2
3	3	4
4	4	1
5	5	4
6	6	1
7	7	2
8	8	3
9	9	4
10	10	1
11	11	2
12	12	3

Output 31.1.3. Design with Lower A-Efficiency

Obs	month	habitat
1	1	4
2	2	2
3	3	3
4	4	1
5	5	1
6	6	4
7	7	4
8	8	1
9	9	2
10	10	1
11	11	4
12	12	3

Note the structure of the best design in Output 31.1.2. One habitat is sampled in each month, each habitat is sampled three times, and the habitats are sampled in consecutive complete blocks. Even though the design in Output 31.1.3 is as D-efficient as the best, it has almost none of this structure; one habitat is sampled each month, but habitats are not sampled an equal number of times. This demonstrates the importance of choosing a final design on the basis of more than one criterion.

You can try searching for the A-optimal design directly. This takes more time but (with only 48 candidate points) is not too large a problem. The following statements produce Output 31.1.4:

```
proc optex seed=193030034 data=a;
   class    habitat;
   model    habitat month c1-c4 s1-s3 / noint;
   generate n=12 criterion=A;
run;
```

Output 31.1.4. Searching Directly for an A-efficient Design

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	31.6103	19.7379	57.7350	1.3229
2	30.1304	17.8273	52.2233	1.3894
3	30.1304	17.7943	52.2233	1.3944
4	30.1304	17.6471	52.2233	1.4093
5	28.1616	15.7055	44.7214	1.4860
6	28.1616	14.5289	44.7214	1.5343
7	28.1616	13.8603	39.2232	1.5811
8	25.0891	11.6152	37.7964	1.8143
9	25.0891	10.7563	37.7964	1.8143
10	25.0891	10.5437	33.3333	1.8930

The best design found is no more A-efficient than the one found previously.

Example 31.2. Comparing DETMAX Algorithm to Sequential Algorithm

See OPTEX4 in the SAS/QC Sample Library

An automotive engineer wants to fit a quadratic model to fuel consumption data in order to find the values of the control variables that minimize fuel consumption (refer to Vance 1986). The three control variables and their possible settings are shown in the following table:

Variable	Values							
AF	15	16	17	18				
EGR	0.020	0.177	0.377	0.566	0.921	1.117		
SA	10	16	22	28	34	40	46	52

Rather than run all 192 ($4 \times 6 \times 8$) combinations of these factors, the engineer would like to see whether the total number of runs can be reduced to 50 in an optimal fashion.

Since the factors have different numbers of levels, you can use the PLAN procedure (refer to the *SAS/STAT User's Guide*) to generate the full factorial set to serve as a candidate data set for the OPTEX procedure.

```
proc plan;
   factors af=4 ordered egr=6 ordered sa=8 ordered
           / noprint;
   output out=a
          af  nvals=(15,16,17,18)
          egr nvals=(.020,.177,.377,.566,.921,1.117)
          sa  nvals=(10,16,22,28,34,40,46,52);
run;
```

The DETMAX algorithm of Mitchell (1974a) is very commonly used for computer-generated optimal design. Although it is not the default search method for the OPTEX procedure, you can specify that it be used with the METHOD=DETMAX option in the GENERATE statement. For example, the following statements produce Output 31.2.1.

```
proc optex data=a seed=61552;
   model af|egr|sa@2 af*af egr*egr sa*sa;
   generate n=50 method=detmax;
run;
```

Output 31.2.1. Efficiencies with DETMAX Algorithm

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	46.4922	24.8987	95.2281	0.4202
2	46.4864	24.8562	95.5744	0.4205
3	46.4797	24.8830	95.3137	0.4203
4	46.4635	25.6461	94.8125	0.4175
5	46.4495	24.5376	95.5559	0.4237
6	46.4459	25.0749	94.8536	0.4197
7	46.4428	24.5111	95.3704	0.4240
8	46.4333	25.0321	95.1371	0.4199
9	46.4333	25.0321	95.1371	0.4199
10	46.4333	25.0321	95.1371	0.4199

The DETMAX search method can require considerable run time. For comparison, you can use the METHOD=SEQUENTIAL option in the GENERATE statement, as shown in the following statements, which produce Output 31.2.2.

```
proc optex data=a seed=33805;
   model af|egr|sa@2 af*af egr*egr sa*sa;
   generate n=50 method=sequential;
run;
```

Output 31.2.2. Efficiencies with Sequential Algorithm

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	46.4009	25.0472	93.8673	0.4200

In a fraction of the run time required by DETMAX, the sequential algorithm finds a design with a relative D-efficiency of $46.4009/46.4922 = 99.8\%$ compared to the best design found by the DETMAX procedure and with *better* A-efficiency. As this demonstrates, if absolute D-optimality is not required, a faster, simpler search may be sufficient.

Example 31.3. Using an Initial Design to Search an Optimal Design

See OPTEX4 in the SAS/QC Sample Library

This example is a continuation of Example 31.2 on page 910.

You can customize the runs used to initialize the search in the OPTEX procedure. For example, you can use the INITDESIGN=SEQUENTIAL option to use an initial design chosen by the sequential search. Or you can place specific points in a data set and use the INITDESIGN=*SAS-data-set* option. In both cases, the search time can be significantly reduced, since the search only has to be done once. This example illustrates both of these options.

The previous example compared the results of the DETMAX and sequential search algorithms. You can use the design chosen by the sequential search as the *starting point* for the DETMAX algorithm. The following statements specify the DETMAX search method, replacing the default initialization method with the sequential search:

```
proc optex data=a seed=33805;
   model af|egr|sa@2 af*af egr*egr sa*sa;
   generate n=50 method=detmax initdesign=sequential;
run;
```

The results, which are displayed in Output 31.3.1, show an improvement over the sequential design itself (Output 31.2.2) but not over the DETMAX algorithm with the default initialization method (Output 31.2.1). Evidently the sequential design represents a local optimum that is not the global optimum, which is a common phenomenon in combinatorial optimization problems such as this one.

Output 31.3.1. Initializing with a Sequential Design

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	46.4333	25.0321	95.1371	0.4199

Prior knowledge of the design problem at hand may also provide a specific set of factor combinations to use as the initial design. For example, many D-optimal designs are composed of replications of the optimal saturated design—that is, the optimal design with exactly as many points as there are parameters to be estimated. In this case, there are 10 parameters in the model. Thus, you can find the optimal saturated design in 10 points, replicate it five times, and use the resulting design as an initial design, as follows:

```
proc optex data=a seed=33805;
    model af|egr|sa@2
          af*af egr*egr sa*sa;
    generate n=saturated
             method=detmax;
    output out=b;


data c; set b; drop i;
   do i=1 to 5; output; end;

proc optex data=a seed=33805;
    model af|egr|sa@2
          af*af egr*egr sa*sa;
    generate n=50
             method=detmax
             initdesign=c;
run;
```

The results are displayed in Output 31.3.2 and Output 31.3.3. The resulting design is 99.9% D-efficient and 98.4% A-efficient relative to the best design found by the straight-forward approach (Output 31.2.1), and it takes considerably less time to produce.

Output 31.3.2. Efficiencies for the Unreplicated Saturated Design

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	41.6990	24.8480	67.6907	0.9508
2	41.4931	22.2840	70.8532	0.9841
3	40.9248	20.7672	62.2177	1.0247
4	40.7447	21.6253	52.7537	1.0503
5	39.9563	20.1557	46.4244	1.0868
6	39.9287	19.5856	45.9023	1.0841
7	39.9287	19.5856	45.9023	1.0841
8	38.9078	13.5976	37.7964	1.2559
9	38.9078	13.5976	37.7964	1.2559
10	37.6832	12.5540	45.3315	1.3036

Output 31.3.3. Initializing with a Data Set

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	46.4388	24.4951	96.0717	0.4242

Example 31.4. Optimal Design Using an Augmented Best Design

See OPTEX4 in the SAS/QC Sample Library

This example is a continuation of Example 31.2 on page 910.

You can specify a set of points that you want included in the final design found by the OPTEX procedure, using the AUGMENT= option in the GENERATE statement to specify a data set that contains a design to be augmented.

In this case, you can try to speed up the search for a 50-run design by first finding an optimal 25-run design and then augmenting that design with another 25 runs, as shown in the following statements:

```
proc optex data=a seed=36926;
    model af|egr|sa@2 af*af egr*egr sa*sa;
    generate n=25 method=detmax;
    output out=b;
proc optex data=a seed=37034;
    model af|egr|sa@2 af*af egr*egr sa*sa;
    generate n=50 method=detmax augment=b;
run;
```

The result (see Output 31.4.1 and Output 31.4.2) is a design with almost 100% D-efficiency and A-efficiency relative to the best design found by the first attempt. However, this approach is not much faster than the original approach, since the run time for the DETMAX algorithm is essentially linear in the size of the design (see "Memory and Run-Time Considerations" on page 942.)

Output 31.4.1. Efficiencies for the 25-point Design to be Augmented

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	46.2975	26.0374	91.1822	0.5849
2	46.2171	25.9733	86.4608	0.5859
3	46.1720	25.9378	88.3293	0.5860
4	46.1374	25.9128	86.1895	0.5866
5	46.0808	22.6647	86.1502	0.6169
6	46.0620	24.7326	89.7179	0.6012
7	45.9992	25.4549	90.3330	0.5946
8	45.9630	24.7610	88.2701	0.5991
9	45.9627	25.5310	88.5737	0.5894
10	45.7994	24.5645	87.7544	0.6005

Output 31.4.2. Efficiencies for the Augmented 50-point Design

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	46.4957	25.0858	94.8160	0.4195
2	46.4773	25.0696	95.0646	0.4195
3	46.4684	24.5519	96.1259	0.4234
4	46.4676	24.5002	95.6830	0.4238
5	46.4587	25.0709	94.6650	0.4196
6	46.4555	24.8087	95.7768	0.4209
7	46.4471	24.5460	95.0073	0.4240
8	46.4373	25.0740	94.4640	0.4194
9	46.3899	25.0007	95.2162	0.4201
10	46.3662	24.4013	94.9539	0.4242

Example 31.5. Optimal Design Using a Small Candidate Set

This example is a continuation of Example 31.4 on page 914.

See OPTEX4 in the SAS/QC Sample Library

A well-chosen initial design can speed up the search procedure, as illustrated in Example 31.2 on page 910. Another way to speed up the search is to reduce the candidate set. The following statements generate the optimal design with a fast, sequential search and then use the FREQ procedure to examine the frequency of different factor levels in the final design:

```
proc optex data=a seed=33805 noprint;
   model af|egr|sa@2 af*af egr*egr sa*sa;
   generate n=50 method=sequential;
   output out=b;
proc freq;
   table af egr sa / nocum;
run;
```

Output 31.5.1. Factor Level Frequencies for Sequential Design

The FREQ Procedure

af	Frequency	Percent
15	19	38.00
16	6	12.00
17	6	12.00
18	19	38.00

egr	Frequency	Percent
0.02	20	40.00
0.566	9	18.00
1.117	21	42.00

sa	Frequency	Percent
10	20	40.00
28	5	10.00
34	6	12.00
52	19	38.00

From Output 31.5.1, it is evident that most of the factor values lie in the middle or at the extremes of their respective ranges. This suggests looking for an optimal design with a candidate set that includes only those points in which the factors have values in the middle or at the extremes of their respective ranges. The following statements illustrate this approach (see Output 31.5.2):

```
proc plan;
   factors af=4 ordered egr=4 ordered sa=4 ordered
           / noprint;
   output out=a af  nvals=(  15,  16,  17,   18)
                egr nvals=(.020,.377,.566,1.117)
                sa  nvals=(  10,  28,  34,   52);
proc optex seed=61552;
   model af|egr|sa@2 af*af egr*egr sa*sa;
   generate n=50 method=detmax;
run;
```

Output 31.5.2. Optimal Design Using a Smaller Candidate Set

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	46.5151	24.9003	96.7226	0.4442
2	46.4997	24.5549	96.1157	0.4478
3	46.4920	24.5530	95.9941	0.4480
4	46.4657	24.8653	95.5627	0.4446
5	46.4547	24.5071	96.0385	0.4481
6	46.4333	25.0321	95.1371	0.4448
7	46.4333	25.0321	95.1371	0.4448
8	46.4333	25.0321	95.1371	0.4448
9	46.3916	24.3617	95.0041	0.4489
10	46.3379	24.8695	94.3115	0.4458

Once again, the resulting design is almost as good as the best one derived by a straightforward search ($> 99.9\%$ relative D-efficiency and $> 98.5\%$ relative A-efficiency) and takes much less time to find. Moreover, designs with fewer factor levels can be much easier to implement.

See "Handling Many Variables" on page 883 for another example of reducing the candidate set for the optimal design search.

Example 31.6. Bayesian Optimal Design

See OPTEX7 in the SAS/QC Sample Library

Suppose you want a design in 20 runs for seven two-level factors. There are 29 terms in a full second-order model, so you will not be able to estimate all main effects and two-factor interactions. If the number of runs were a power of 2, a design of resolution 4 could be used to estimate all main effects free of the two-factor interactions, as well as to provide partial information on the interactions. However, when the number

of runs is not a power of two, as in this case, DuMouchel and Jones (1994) suggest searching for a *Bayesian optimal design* by specifying nonzero prior precision values for the interactions. You can specify these values in the OPTEX procedure with the PRIOR= option in the MODEL statement. This says that you want to consider all main effects and interactions as potential effects, but you are willing to sacrifice information on the interactions to obtain maximal information on the main effects. When an orthogonal design of resolution 4 exists, it is optimal according to this Bayesian criterion. You can use the following statements to generate the Bayesian D-optimal design:

```
proc factex;
   factors x1-x7;
   output out=can;
run;

proc optex data=can seed=57922
           coding=orth;
   model x1-x7,
         x1|x2|x3|x4|x5|x6|x7@2
         / prior=0,16;
   generate n=20 method=m_fedorov;
   output out=des;
run;
```

With orthogonal coding, the value of the prior for an effect says roughly how many prior "observations' worth" of information you have for that effect. In this case, the PRIOR= precision values and the use of commas to group effects in the MODEL statement says that there is no prior information for the main effects and 16 runs' worth of information for each two-factor interaction. See "Design Coding" on page 937 for details on orthogonal coding.

The efficiencies are shown in Output 31.6.1.

Output 31.6.1. Efficiencies for Bayesian Optimal Designs

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	85.1815	74.6705	85.2579	1.1476
2	85.1815	74.6705	85.2579	1.1476
3	85.1815	74.6705	85.2579	1.1476
4	85.0424	73.3109	81.0800	1.1582
5	85.0424	73.3109	81.0800	1.1582
6	84.5680	73.5053	84.1376	1.1566
7	84.5480	74.1000	85.9714	1.1520
8	84.4931	72.1671	81.7855	1.1673
9	84.4239	72.4979	81.7431	1.1646
10	84.3919	74.6097	89.3631	1.1480

Notice that the best design was found in three tries out of ten. It may be a good idea to repeat the search with more tries (see the ITER= option on page 903.) You can use the ALIASING option of the GLM procedure to list the aliasing structure for the design:

```
data des; set des;
   y = ranuni(654231);
proc glm data=des;
   model
      y = x1-x7
          x1|x2|x3|x4|x5|x6|x7@2
          / e aliasing;
run;
```

The relevant part of the output is shown in Output 31.6.2. Most of the main effects are indeed unconfounded with two-factor interactions, although many two-factor interactions are confounded with each other.

Output 31.6.2. Aliasing Structure for Bayesian Optimal Design

```
                          The GLM Procedure

                  General Form of Aliasing Structure

                  Intercept
                  x1 - 0.5*x3*x7
                  x2
                  x3
                  x4 + 0.5*x3*x7
                  x5
                  x6
                  x7
                  x1*x2 - x3*x6 + 0.5*x3*x7 - x4*x7
                  x1*x3 - x2*x6 - x5*x7
                  x2*x3 + x3*x7
                  x1*x4 - x5*x6 + x5*x7 + x6*x7
                  x2*x4 - x3*x6 + 0.5*x3*x7 - x4*x7
                  x3*x4 - x2*x6 - x5*x7
                  x1*x5 - x4*x6 - x3*x7
                  x2*x5 + x2*x6 + x5*x7 + x6*x7
                  x3*x5 + x3*x6 - x3*x7
                  x4*x5 - x1*x6 - x3*x7
                  x1*x7 - x4*x7
                  x2*x7 + x5*x7 + x6*x7



                          The GLM Procedure

Dependent Variable: y

NOTE: The X'X matrix has been found to be singular, and a generalized
      inverse was used to solve the normal equations.  Terms whose
      estimates are followed by the letter 'B' are not uniquely estimable.
```

Example 31.7. Balanced Incomplete Block Design

See OPTEX8 in the SAS/QC Sample Library

This example uses the BLOCKS statement to construct an incomplete block design. An incomplete block design is a design for v qualitative treatments in b blocks of k runs each, where $k < v$ so that not all treatments can occur in each block. An incomplete block design is said to be *balanced* when all pairs of treatments occur equally often in the same block. A balanced design is always optimal for any criterion based on the information matrix, although there are many values of (v, b, k) for which no balanced design exists.

One way to construct an incomplete block design with the OPTEX procedure is to include the blocking factor in the candidate set and in the model. For example, the following statements search for a BIBD for seven treatments in seven blocks of size three—that is, $(v, b, k) = (7, 7, 3)$—using the full set of 49 treatment-by-block combinations for candidates:

```
data can;
   do tmt = 1 to 7;
      do blk = 1 to 7;
         output;
         end;
      end;

proc optex data=can seed=8327
           coding=orth;
   class tmt blk;
   model tmt blk;
   generate n=21;
run;
```

By default, the OPTEX procedure performs the search 10 times from different random starting designs. The various efficiencies for each design are listed in Output 31.7.1.

Output 31.7.1. Efficiency Factors for $v = b = 7$, $k = 3$ Designs

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	89.0483	79.1304	82.7170	0.8845
2	89.0483	79.1304	82.7170	0.8845
3	88.4669	76.9882	78.6796	0.8967
4	88.4669	76.9882	78.6796	0.8967
5	88.4669	76.9882	78.6796	0.8967
6	88.4669	76.9882	78.6796	0.8967
7	88.4669	76.9882	78.6796	0.8967
8	88.4669	76.9882	78.6796	0.8967
9	88.1870	76.0262	78.7612	0.9024
10	87.7681	74.2459	73.9544	0.9131

Since the efficiency factors compare the designs to a (hypothetical) orthogonal design, values of 100% are not possible in this case. The OPTEX procedure includes facilities for examining the information matrix for the design; you can use these to verify that the best design found here is, in fact, balanced.

Searching for an optimal design for both treatments and blocks simultaneously has its limitations. Note that the balanced design was found on only two of the ten tries. A more serious limitation is that this approach sometimes fails to find a design with equal-sized blocks. A more efficient and flexible way to construct a block design with the OPTEX procedure is to use the BLOCKS statement.

The following statements use the BLOCKS statement to solve the incomplete block design problem described previously. In this case, the candidate set simply consists of the seven treatment levels.

```
data can;
   do tmt = 1 to 7;
      output;
      end;
proc optex data=can seed=73462
           coding=orth;
   class tmt;
   model tmt;
   blocks structure=(7)3;
run;
```

The output again consists of efficiency factors for 10 different tries, but this time the factors are computed from the information matrix for only the treatment effects. In this special case (a single classification effect in the treatment model together with the BLOCKS STRUCTURE= specification), the efficiency of each design as an incomplete block design is also listed (Output 31.7.2).

Output 31.7.2. Efficiency Factors for $v = b = 7$, $k = 3$ Optimal Blocking Designs

The OPTEX Procedure

Design Number	Treatment D-Efficiency	Treatment A-Efficiency	Block Design D-Efficiency
1	77.7778	77.7778	100.0000
2	77.7778	77.7778	100.0000
3	77.7778	77.7778	100.0000
4	77.7778	77.7778	100.0000
5	77.7778	77.7778	100.0000
6	77.7778	77.7778	100.0000
7	77.7778	77.7778	100.0000
8	77.7778	77.7778	100.0000
9	77.7778	77.7778	100.0000
10	77.7778	77.7778	100.0000

The 100% efficiency in the fourth column of the output shows that the balanced design was found on all 10 tries.

Since the OPTEX procedure is interactive, you can save the final design in a data set by submitting the OUTPUT statement immediately after the preceding statements. The following statements use the BLOCKNAME= option to rename the block variable:

```
   output out=bibd blockname=blk;
proc print data=bibd;
run;
```

The final design is shown in Output 31.7.3.

Although there is no guarantee that the OPTEX procedure will find the globally optimal block design by this method, it usually does find small- to medium-sized balanced designs, and it always finds a very efficient design. For example, for the designs given in Table 9.5 of Cochran and Cox (1957), the OPTEX procedure consistently finds the theoretically optimal BIBD in all cases with 10 or fewer treatments. Furthermore, in no case is the D-efficiency relative to the balanced design less than 99%.

Output 31.7.3. Balanced Incomplete Block Design for $v = b = 7$, $k = 3$

Obs	BLK	tmt
1	1	3
2	1	4
3	1	7
4	2	6
5	2	3
6	2	5
7	3	2
8	3	3
9	3	1
10	4	7
11	4	1
12	4	6
13	5	5
14	5	4
15	5	1
16	6	5
17	6	7
18	6	2
19	7	4
20	7	6
21	7	2

Example 31.8. Optimal Design with Fixed Covariates

See OPTEX9 in the SAS/QC Sample Library

In addition to finding optimal block designs, you can use the BLOCKS statement to find designs that are optimal with respect to more general covariate models. You can specify the data set containing the covariates with the DESIGN= option in the BLOCKS statement. Covariate models are specified in the same way as the treatment model.

The following example is based on an example in Harville (1974). Suppose you want a design for five qualitative treatments in 10 runs. The value of a covariate thought

to be related to the response has been recorded for each of the experimental units. For instance, if the treatments are different types of animal feed, a typical covariate might be the initial weight of each animal. In the following, the data sets COV and TMT are created, containing the covariate values and the candidate treatment levels, respectively. Then the OPTEX procedure is invoked with a simple one-way model for the treatment effect and a quadratic model for the covariate effect.

```
data cov; input u @@; datalines;
.46 .54 .58 .60 .73 .77 .82 .84 .89 .95
;
data tmt;
   do t = 1 to 5;
      output;
      end;
proc optex data=tmt seed=17364
           coding=orthcan;
   class t;
   model t;
   blocks design=cov;
   model u u*u;
   output out=tmtu;
proc print data=tmtu;
run;
```

In this case, the CODING=ORTHCAN option in the PROC OPTEX statement has the same effect as CODING=ORTH, which is to produce orthogonal coding with respect to the candidates. Note that

- the CLASS and MODEL statements that define the treatment model precede the BLOCKS statement
- the MODEL statement that defines the covariate model follows the BLOCKS statement

As a general rule, CLASS and MODEL statements that come before a BLOCKS statement are interpreted as applying to the treatment model, while CLASS and MODEL statements that come after a BLOCKS statement involving the DESIGN= blocks-specification are interpreted as applying to the covariate model.

The listing of the efficiency values for the 10 designs found is shown in Output 31.8.1. Note that the efficiencies are the same for all tries. A listing of the design is shown in Output 31.8.2.

Output 31.8.1. Optimal Treatment Efficiency Factors with a Quadratic Covariate Effect

The OPTEX Procedure

Design Number	Treatment D-Efficiency	Treatment A-Efficiency
1	91.6621	91.1336
2	91.6621	91.1336
3	91.6621	91.1336
4	91.6621	91.1336
5	91.6621	91.1336
6	91.6621	91.1336
7	91.6621	91.1336
8	91.6621	91.1336
9	91.6621	91.1336
10	91.6621	91.1336

Output 31.8.2. Optimal Design with a Quadratic Covariate Effect

Obs	u	t
1	0.46	4
2	0.54	3
3	0.58	1
4	0.60	2
5	0.73	5
6	0.77	4
7	0.82	3
8	0.84	1
9	0.89	2
10	0.95	5

When you use the BLOCKS statement without specifying the GENERATE statement, the full candidate set is used as the treatment set for optimal blocking. If you specify both statements, an optimal design for the treatments ignoring the blocks is first generated, and the result is used as the treatment set for optimal blocking. This allows several options to be combined to evaluate existing designs. For example, the following statements evaluate the optimal design given in Harville (1974) for the preceding situation:

```
data har; input t @@; datalines;
1   2   3   4   5   1   2   3   4   5
;
proc optex data=tmt coding=orthcan;
   class t;
   model t;
   generate initdesign=har
            method=sequential;
   blocks design=cov init=chain iter=0;
   model u u*u;
run;
```

The efficiency values for Harville's design are shown in Output 31.8.3. They are the same as for the design found by the OPTEX procedure.

Output 31.8.3. Treatment Efficiency Factors for Harville's Design

The OPTEX Procedure

Design Number	Treatment D-Efficiency	Treatment A-Efficiency
1	91.6621	91.1336

In fact, the optimal design found by OPTEX can be derived from Harville's design simply by re-labeling treatments. In order of increasing U, both designs consist of two consecutive replicates of the treatments, with treatments in both replicates occurring in the same order.

Example 31.9. Optimal Design in the Presence of Covariance

See OPTEX10 in the SAS/QC Sample Library

The BLOCKS statement finds a design that maximizes the determinant $|X'AX|$ of the treatment information matrix, where A depends on the block or covariate model. Alternatively, you can directly specify the matrix A to find the D-optimal design when A is the variance-covariance matrix for the runs. You can specify the data set containing the covariance matrix with the COVAR= option in the BLOCKS statement, listing the variables corresponding to the columns of the covariance matrix in the VAR= option. If you specify n variables in the VAR= option, the values of these variables in the first n observations in the data set will be used to define A.

For example, suppose you want to compare the effects of seven different fertilizers on crop yield, using seven long, narrow blocks of four plots each, as depicted in Figure 31.1 on page 924.

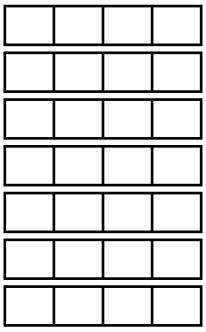


Figure 31.1. Block Structure for Neighbor Balance

In this case, it is reasonable to conjecture that closer plots within each block are more correlated. In particular, suppose that the plots are *autocorrelated*, so that the

correlation matrix for the four plots in each block is of the form

$$R = \begin{bmatrix} 1 & \rho & \rho^2 & \rho^3 \\ \rho & 1 & \rho & \rho^2 \\ \rho^2 & \rho & 1 & \rho \\ \rho^3 & \rho^2 & \rho & 1 \end{bmatrix}$$

where $-1 \leq \rho \leq 1$. If there is also an overall fixed effect due to blocks, the information matrix for the effect of fertilizer has the form $X'AX$, where

$$A = \left(V^{-1} - V^{-1}Z\left(Z'V^{-1}Z\right)^{-1}Z'V^{-1}\right)^{-}$$

In this formula, V is the block diagonal matrix of the plot-by-plot correlation structure, with seven copies of R_4 on the diagonal. The matrix Z is the design matrix corresponding to the block effect. The optimal design should take into account this neighbor covariance structure as well as the block structure.

The following code uses the SAS/IML matrix language to construct A using $\rho = 0.1$ and saves it in a data set named A:

```
proc iml;
   blks = int(((1:28)`-1)/4) + 1;
   z = j(28,1) || designf(blks);

   r = toeplitz(0.1**(0:3));
   v = r;
   do i = 2 to 7; v = block(v,r); end;

   iv = inv(v);
   a = ginv(iv-iv*z*inv(z`*iv*z)*z`*iv);
   create A from a;
   append from a;
quit;
```

Note that the data set is created with variables named COL1, COL2, ..., COL28, by default.

To find an allocation of fertilizers to plots that is optimal for detecting the fertilizer effect in the presence of this autocorrelation, simply specify a one-way model for the treatment effects and specify the data set A as the covariance matrix for the runs with the COVAR= option in the BLOCKS statement, as follows:

```
data fert; do f = 1 to 7; output; end;
proc optex data=fert seed=56672 coding=orth;
   class f;
   model f;
   blocks covar=A
          var=(col1-col28);
   output out=nbd;
run;
```

The SAS/IML matrix language also provides a convenient way of listing the design.

```
proc iml;
   use nbd;                                  Read in the selected levels
   read all var {f};                            of fertilizer
   nbd = shape(f,7,4);                       Reshape them into 7 4-run
   print nbd [format=2.];                       blocks and print.
```

The resulting design is shown in Output 31.9.1. Note that it is not only a balanced incomplete block design, but it is also balanced for first neighbors; that is, every pair of treatments occur equally often on horizontally adjacent plots.

Output 31.9.1. Neighbor-Balanced BIBD for $v = b = 7$, $k = 4$, Found by Optimal Blocking

NBD

7	2	1	5
6	1	7	3
4	7	6	2
1	4	6	5
6	3	5	2
1	3	2	4
7	5	4	3

Example 31.10. Adding Space-Filling Points to a Design

See OPTEX11 in the SAS/QC Sample Library

Suppose you want a 15-run experiment for three mixture factors X1, X2, and X3; furthermore, suppose that X3 cannot account for any more than 75% of the mixture. The vertices and generalized edge centroids of the region defined by these constraints comprise a good candidate set to use with the OPTEX procedure for finding a D-optimal design for such an experiment. However, information-based criteria such as D- and A-efficiency tend to push the design to the edges of the candidate space, leaving large portions of the interior relatively uncovered. For this reason, it is often a good idea to augment a D-optimal design with some points chosen according to U-optimality, which seeks to cover the candidate region as well as possible.

The following statements create a data set containing the vertices and generalized edge centroids of the region defined by the constraints on the factors and plot the candidate set:

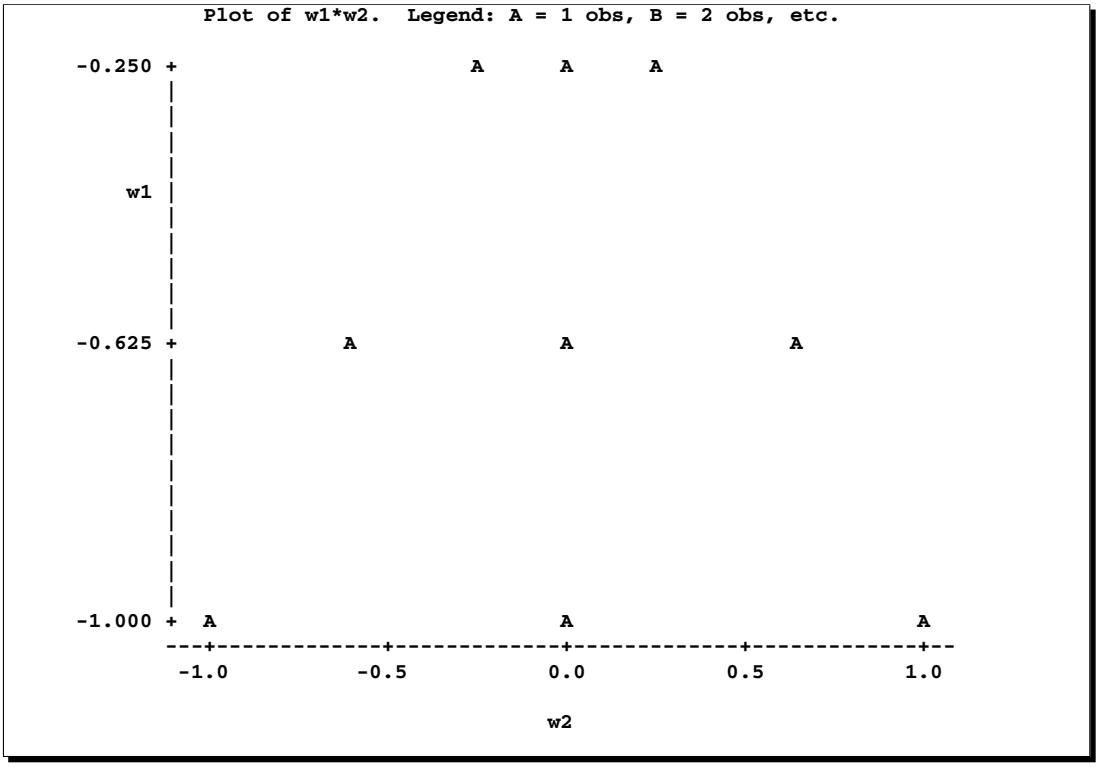
```
data a;
   input x1 x2 x3;
datalines;
1.0000 0.0000 0.000
0.0000 1.0000 0.000
0.0000 0.2500 0.750
0.2500 0.0000 0.750
0.0000 0.6250 0.375
0.6250 0.0000 0.375
```

```
0.5000 0.5000 0.000
0.1250 0.1250 0.750
0.3125 0.3125 0.375
;
data a; set a;
   w1 = -(x1 + x2);
   w2 =  (x1 - x2);
proc plot data=a;
   plot w1*w2;
run;
```

The constraint that the factor levels sum to 1 means that the candidate points all lie on a plane. The transformed variables W1 and W2 are the coordinates of each candidate point with respect to two orthogonal axes in that plane.

The result, shown in Output 31.10.1, is a "quick-and-dirty" plot of the vertices, the edge centroids, and the over-all centroid for the feasible region. The $X3 \leq 0.75$ constraint effectively "cuts off" the top of the usual simplex.

Output 31.10.1. Vertices and Centroids for Constrained Mixture Design



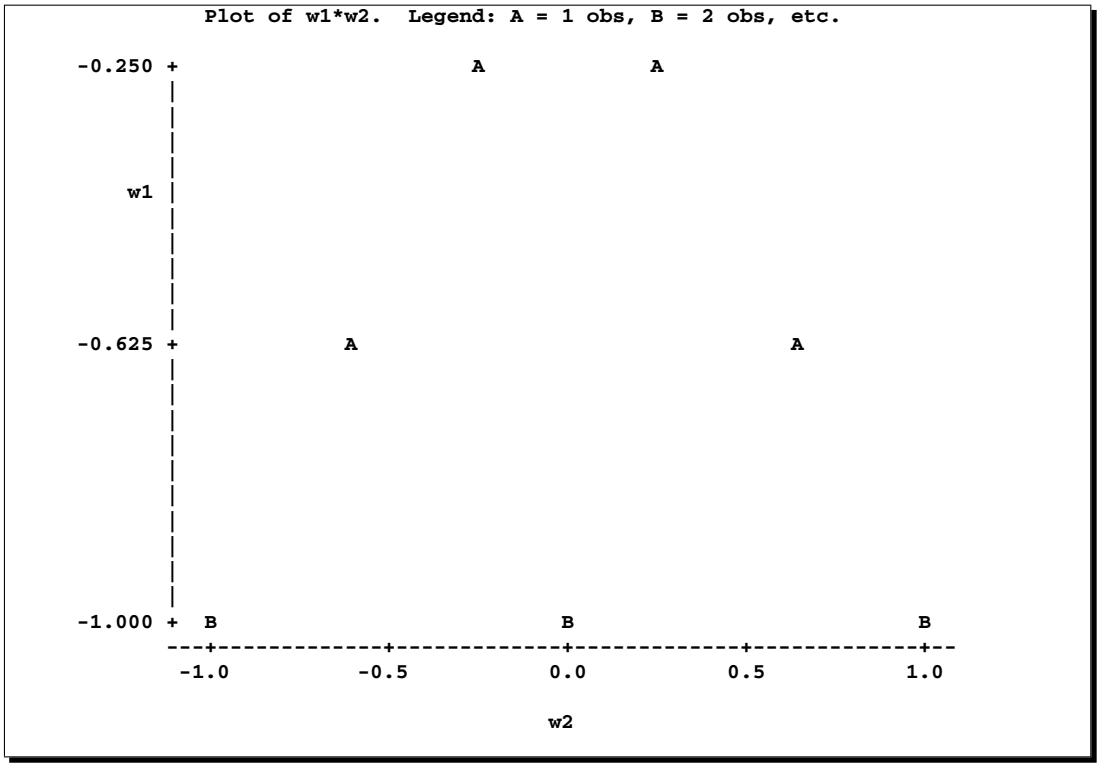
You could easily use this plot to choose 15 runs both to span the extremes of the candidate region and to cover the interior. However, you can use the methods discussed in this section with higher dimensional problems that are difficult or impossible to visualize.

You can use the OPTEX procedure to select 10 optimal points for estimating a second-order model in the mixture factors.

```
proc optex data=a seed=60868 nocode;
   model x1|x2|x3@2 / noint;
   generate n=10;
   output out=b;
data b; set b;
   w1 = -(x1 + x2);
   w2 =  (x1 - x2);
proc plot data=b;
   plot w1*w2;
run;
```

As shown in Output 31.10.2, the D-optimal design omits some of the candidate points and replicates others.

Output 31.10.2. D-optimal Constrained Mixture Design



The D-optimal design leaves large “holes” in the feasible region. The following statements use the SQL procedure to augment the candidate set with the average of every pair of points in it, effectively producing a set of points scattered throughout the feasible region:

```
proc sql;
   create table fill as select distinct
      (one.x1 + two.x1)/2 as x1,
      (one.x2 + two.x2)/2 as x2,
      (one.x3 + two.x3)/2 as x3 from a one, a two;
data a;
   set a fill;
run;

data a; set a;
   w1 = -(x1 + x2);
   w2 =  (x1 - x2);
proc plot data=a;
   plot w1*w2;
run;
```

The results are shown in Output 31.10.3.

Output 31.10.3. Filled Candidate Region for Constrained Mixture Design

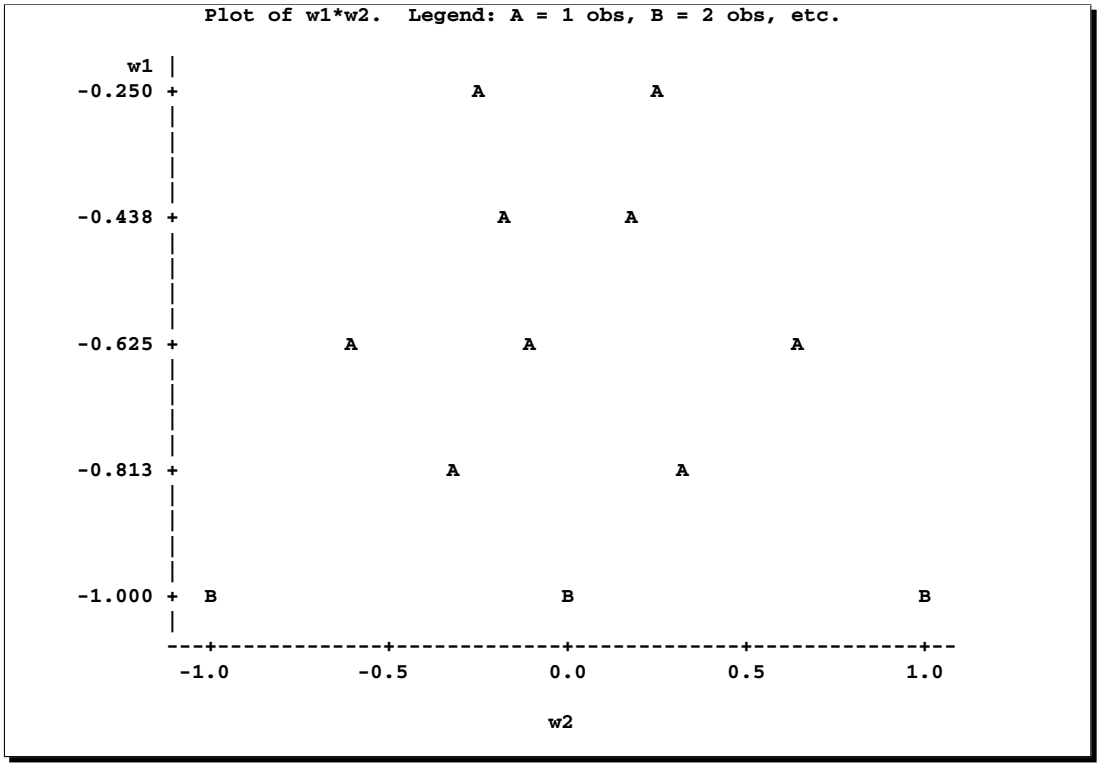
```
            Plot of w1*w2.  Legend: A = 1 obs, B = 2 obs, etc.

     w1 |
-0.250 +                      A   A  A   A  A
        |
        |
        |
        |
-0.438 +                  A  A   A A  A   AA   A  A
        |
        |
        |
        |
-0.625 +             A  A   AA     A  A   A    A A  A   A
        |
        |
        |
        |
-0.813 +       A        A    A   A    A    A   A    A        A
        |
        |
        |
        |
-1.000 +  A             A             A             A             A
        |
        ---+-------------+-------------+-------------+-------------+--
         -1.0          -0.5           0.0           0.5           1.0

                                      w2
```

The filled-in data set A has too many points (recall that the goal is a design with 15 runs), but you can use the OPTEX procedure to choose points from it. The following statements “fill in the holes” in the optimal design saved in B by augmenting it with points chosen from the filled-in data set A to optimize the U-criterion:

```
proc optex data=a seed=4321 nocode;
   model x1 x2 x3 / noint;
   generate n=15 augment=b
            criterion=u;
   output out=c;
data c; set c;
   w1 = -(x1 + x2);
   w2 =  (x1 - x2);
proc plot data=c;
   plot w1*w2;
run;
```

Output 31.10.4 shows that the U-optimal design fills in the candidate region in much the same way that you might construct the design by visually assigning points. That is, the general approach using the OPTEX procedure agrees with visual intuition for this small problem. Moreover, the general approach yields an appropriate design for higher dimensional problems that cannot be visualized.

Output 31.10.4. D-optimal Constrained Mixture Design Filled In U-optimally



Data Details

Input Data Sets

This section discusses the five input data sets for the OPTEX procedure. Three of the data sets provide points used to generate the design according to the effects you specify in the MODEL statement. Two other data sets provide points used to generate a model for fixed covariates.

Only the DATA= data set is required. If you do not specify a DATA= data set in the PROC OPTEX statement, the procedure uses the last data set created as a set of candidate points for the design. The AUGMENT= data set is optional and contains points that must be included in the final design. The INITDESIGN= data set is also optional and provides an initial design to be used by a search procedure. Variables listed in the MODEL statement must be present in all three of these data sets, and the variable characteristics (type and length) must match across data sets.

Figure 31.2 is a schematic diagram of the roles of the DATA=, AUGMENT=, and INITDESIGN= data sets in constructing the design. Figure 31.3 presents the role of the DESIGN= data set for block designs.

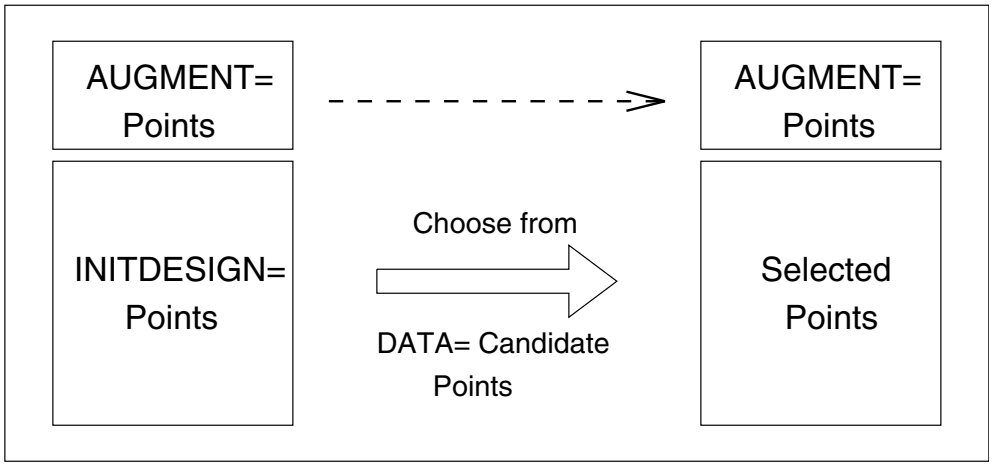


Figure 31.2. Choosing from DATA= Points

DATA= Data Set

The DATA= data set provides a set of candidate points used to create a design. The OPTEX procedure uses the variables listed in the MODEL statement when creating a design.

The effects specified in a MODEL statement determine the variables used when generating a design. For example, if the DATA= data set contains the variables A, B, and C, but the MODEL statement specifies effects involving only A and B, then the variable C is not considered when generating designs.

Variables in the DATA= data set that are listed in the ID statement are transferred to the OUT= data set (if one is created).

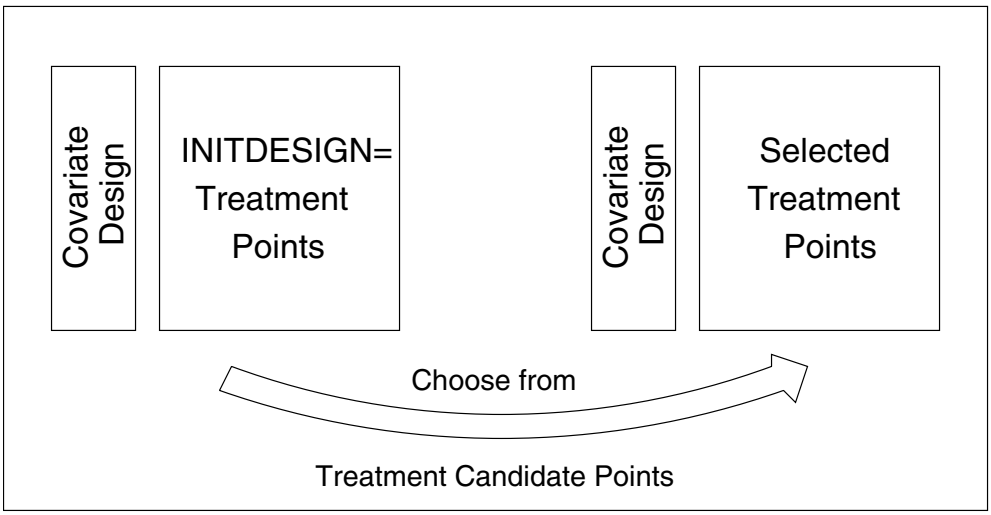


Figure 31.3. Choosing Treatment Candidates

AUGMENT= Data Set

The AUGMENT= data set provides a set of points that must be included in the final design. The OPTEX procedure adds candidate points from the DATA= data set to the points from the AUGMENT= data set when generating designs. The number of points in the AUGMENT= data set must be less than or equal to the number of points for the design (either the default or the number specified with the N= option in the GENERATE statement).

As with the DATA= data set, the effects specified in a MODEL statement determine the variables used when generating a design. The types and lengths of variables in an AUGMENT= data set that are used in the MODEL and ID statements must match the types and lengths of the same variables in the DATA= data set. If you use an ID statement and the AUGMENT= data set contains the ID variables, these variables are transferred to the OUT= data set (if one is created). See "Including Specific Runs" on page 880 for an example that uses an AUGMENT= data set.

INITDESIGN= Data Set

The INITDESIGN= data set provides a set of points that are used as an initial design in the search for an optimal design. These points are not necessarily contained in the final design. The OPTEX procedure uses these points to begin the search for an optimal design. The number of points in the INITDESIGN= data set must be the same as the number of points in the design (either the default or the number specified with the N= option in the GENERATE statement).

As with the DATA= data set, the effects specified in a MODEL statement determine the variables used when generating a design. The types and lengths of variables in an INITDESIGN= data set that are used in the MODEL and ID statements must match the types and lengths of the same variables in the DATA= data set. If you use an ID statement and the INITDESIGN= data set contains the ID variables, these variables are transferred to the OUT= data set (if one is created). See Example 31.3 on page 912 for an example that uses an INITDESIGN= data set.

If you use an INITDESIGN= data set and also specify METHOD=SEQUENTIAL in the GENERATE statement, no search is performed. The INITDESIGN= data set is the final design. In this way, you can use the OPTEX procedure to evaluate an existing design.

BLOCKS DESIGN= Data Set

The DESIGN= data set in the BLOCKS statement contains a set of points that are used to generate a model for fixed covariates. These points are contained in the final design and are transferred to the OUT= data set (if one is created). See Example 31.8 on page 921 for an example that uses a BLOCKS DESIGN= data set.

BLOCKS COVAR= Data Set

If you specify a COVAR= data set in the BLOCKS statement, the observations for the variables listed in the VAR= option are used to define the assumed variance-covariance matrix for the experimental runs. These observations are *not* transferred to the OUT= data set (if one is created). Note that since covariance matrices are necessarily square, the number of observations in the COVAR= data set must be the same as the number of variables listed in the VAR= option. See Example 31.9 on page 924 for an example that uses a BLOCKS COVAR= data set.

Output Data Sets

You typically use the OPTEX procedure to create an output data set that contains the design for your experiment. If you use an OUTPUT statement, the variables in the output data set are the factors of the design as well as any ID variables. The values for the ID variables are taken from the input data set (the DATA=, AUGMENT=, or INITDESIGN= data set) that provided the design point. ID variables must be contained in the DATA= data set and can also be contained in the AUGMENT= or INITDESIGN= data sets. If an AUGMENT= or INITDESIGN= data set does not contain the ID variables, and points from the data set are used in the final design, values of ID variables for those points are missing.

Since the input data sets provide candidate points for the design, all the observations in the OUT= data set originate in one of the input data sets. The OPTEX procedure does not change the values of variables in the input data sets.

Since you can use multiple OUTPUT statements with the OPTEX procedure, you can create multiple OUT= data sets in a given run of the procedure.

Computational Details

Specifying Effects in MODEL Statements

This section discusses how to specify the linear model that you plan to fit with the design. The OPTEX procedure provides for the same general linear models as the GLM procedure, although it does not use the GLM procedure's *over-parameterized* technique for generating the design matrix (see "Static Coding" on page 938.)

Each term in a model, called an *effect*, is a variable or combination of variables. To specify effects, you use a special notation involving variables and operators. There are two kinds of variables: *classification variables* and *continuous variables*. *Classification variables* separate observations into groups, and the model depends on them through these groups; on the other hand, the model depends on the actual (or coded) values of *continuous variables*. There are two primary operators: *crossing* and *nesting*. A third operator, the *bar operator*, simplifies the specification for multiple crossed terms, as in a factorial model. The **@** operator, used in combination with the bar operator, further simplifies specification of crossed terms.

When specifying a model, you must list the classification variables in a CLASS statement. Any variables in the model that are not listed in the CLASS statement are assumed to be continuous. Continuous variables must be numeric.

Types of Effects

Five types of effects can be specified in the MODEL statement. Each row of the design matrix is generated by combining values for the independent variables according to effects specified in the MODEL statement. This section discusses how to specify different types of effects and explains how they relate to the columns of the design matrix. In the following, assume that A, B, and C are classification variables and X1, X2, and X3 are continuous variables.

Regressor Effects

Regressor effects are specified by writing continuous variables by themselves.

```
X1  X2  X3
```

For regressor effects, the actual values of the variable are used in the design matrix.

Polynomial Effects

Polynomial effects are specified by joining two or more continuous variables with asterisks.

```
X1*X1  X1*X1*X1  X1*X2  X1*X2*X3  X1*X1*X2
```

Polynomial effects are also referred to as interactions or cross products of continuous variables; when a variable is joined with itself, polynomial effects are referred to as quadratic effects, cubic effects, and so on. In the preceding examples, the first two effects are the quadratic and cubic effects for X1, respectively. The remaining effects are cross products.

For polynomial effects, the value used in the design matrix is the product of the values of the constituent variables.

Main Effects

If a classification variable A has k levels, then its main effect has $k-1$ degrees of freedom, corresponding to $k-1$ independent differences between the mean response at different levels. Main effects are specified by writing class variables by themselves.

```
A  B  C
```

Most designs involve main effects since these correspond to the factors in your experiment. For example, in a factorial design for a chemical process, the main effects may be temperature, pressure, and the level of a catalyst.

For information on how the OPTEX procedure generates the $k-1$ columns in the design matrix corresponding to the main effect of a classification variable, see "Design Coding" on page 937.

Crossed Effects

Crossed effects (or interactions) are specified by joining class variables with asterisks.

```
A*B  B*C  A*B*C
```

The number of degrees of freedom for a crossed effect is the product of the numbers of degrees of freedom for the constituent main effects. The columns in the design matrix corresponding to a crossed effect are formed by the horizontal direct products of the constituent main effects.

Continuous-by-Class Effects

Continuous-by-class effects are specified by joining continuous variables and class variables with asterisks.

```
X1*A
```

The design columns for a continuous-by-class effect are constructed by multiplying the values in the design columns for the continuous variables and the class variable.

Note that all design matrices start with a column of ones for the assumed intercept term unless you use the NOINT option in the MODEL statement.

Bar and @ Operators

You can shorten the specification of a factorial model using the bar operator. For example, the following statements show two ways of specifying a full three-way factorial model:

```
model a b c a*b a*c b*c a*b*c;
model a|b|c;
```

When the vertical bar (**|**) is used, the right- and left-hand sides become effects, and their cross becomes an effect. Multiple bars are permitted. The expressions are expanded from left to right using rules given by Searle (1971). For example, **A|B|C** is evaluated as follows:

```
A | B | C  →  { A | B } | C
           →  { A B A*B } | C
           →  A  B  A*B  C  A*C  B*C  A*B*C
```

The bar operator does not cross a variable with itself. To produce a quadratic term, you must specify it directly.

You can also specify the maximum number of variables involved in any effect that results from bar evaluation by putting it at the end of a bar effect, preceded by an **@** sign. For example, the specification **A|B|C@2** results in only those effects that contain two or fewer variables (in this case A, B, A*B, C, A*C, and B*C.)

Examples of Models

Main Effects Model

For a three-factor main effects model with A, B, and C as the factors, the MODEL statement is

```
model a b c;
```

Factorial Model with Interactions

To specify interactions in a factorial model, join effects with asterisks, as described previously. For example, the following statements show two ways of specifying a complete factorial model, which includes all the interactions:

```
model a b c a*b a*c b*c a*b*c;
model a|b|c;
```

Quadratic Model

The following statements show two ways of specifying a model with crossed and quadratic effects (for a central composite design, for example):

```
model x1 x2 x1*x2 x3 x1*x3 x2*x3
      x1*x1 x2*x2 x3*x3;
model x1|x2|x3@2 x1*x1 x2*x2 x3*x3;
```

Design Efficiency Measures

The output from the OPTEX procedure includes efficiency measures for the resulting designs according to various criteria. This section gives the precise definitions for these measures.

By default, the OPTEX procedure calculates the following efficiency measures for each design found in its search for an optimum design:

$$\text{D-efficiency} = 100 \times \left(\frac{|X'X|^{1/p}}{N_D} \right)$$

$$\text{A-efficiency} = 100 \times \left(\frac{p/N_D}{\text{trace}(X'X)^{-1}} \right)$$

$$\text{G-efficiency} = 100 \times \left(\sqrt{\frac{p/N_D}{\max_{\mathbf{x} \in \mathcal{C}} \mathbf{x}'(X'X)^{-1}\mathbf{x}}} \right)$$

where p is the number of parameters in the linear model, N_D is the number of design points, and $\mathcal{C}$ is the set of candidate points. The D- and A-efficiencies are the relative number of runs (expressed as percents) required by a hypothetical orthogonal design to achieve the same $|X'X|$ and $\mathrm{trace}(X'X)^{-1}$, respectively; refer to Mitchell (1974b).

When you specify a BLOCKS statement, the D- and A-efficiencies for the treatment part of the model are calculated. These are calculated similarly to the preceding efficiencies, except that they are based on the information matrix after correcting for covariate effects. This matrix can be written as $X'AX$ for a symmetric, positive definite matrix A that depends on the model for the covariate effect. If you specify a block structure or a covariate model, then $A = I - Z(Z'Z)^{-1}Z'$, where Z is the design matrix for the block or covariate effect. Alternatively, you can use the COVAR= option to specify the matrix A directly. Given A, the efficiencies in the presence of covariates are defined as follows:

$$\begin{array}{rcl@{\qquad}rcl} \text{D-efficiency} & = & 100 \times c_D^{-1} \cdot |X'AX|^{1/p}/N, & c_D & = & \prod_{i=1}^{p} \lambda_i^{1/p} \\ \text{A-efficiency} & = & 100 \times c_A \cdot (p/N)/\mathrm{trace}(X'AX)^{-1}, & c_A & = & \sum_{i=1}^{p} \lambda_i/p \end{array}$$

where $\lambda_1, \ldots, \lambda_p$ are the p largest eigenvalues of A. If you use the STRUCTURE= block model specification and there is only one class variable in the treatment model, then the design fits into the traditional block design framework. In this case, the D-efficiency relative to a balanced incomplete block design is also listed.

Because these efficiencies measure the goodness of the design relative to theoretical designs that may be far from possible in many cases, they are typically not useful as absolute measures of design goodness. Instead, efficiency measures should be used relatively, to compare one design to another for the same situation.

For the distance-based criteria, there are no simple measures of design efficiency that can be scaled from 0 to 100. See the "Output" section on page 947 for a definition of the design measures tabulated for these criteria.

Design Coding

The way the independent effects of the model are interpreted to generate a linear model is called *coding*. The OPTEX procedure provides for different types of coding. For D-optimality, the type of coding affects only the absolute value of the computed efficiency criteria, not the relative values for two different designs. Thus, different codings do not affect the choice of D-optimal design. In this section, the details and ramifications of the different types of coding are discussed.

Coding the points in a design involves selecting linearly independent columns corresponding to each model term, turning particular values of the factors into a row vector $\mathbf{x}$. The OPTEX procedure requires a *non-singular* coding for the design matrix. Because of this, any two coding schemes are related by a non-singular transformation.

Static Coding

The default coding for the design points is as follows:

- Unless you specify CODING=NONE (or NOCODE) in the PROC OPTEX statement, continuous variables are centered and scaled so that their maximum and minimum values are 1 and –1, respectively.
- The $k-1$ columns corresponding to the main effect of a classification variable A are computed as follows: For a design point with A at its i^{th} level, for $1 \leq i \leq k-1$, the columns of the design matrix associated with A are all 0 except for the i^{th} column, which is 1. When A is at its k^{th} level, all $k-1$ columns associated with A are -1. Thus, if α_i denotes the expected response at the i^{th} level of A, the $k-1$ columns yield estimates of $\alpha_1 - \alpha_k, \alpha_2 - \alpha_k, \ldots, \alpha_{k-1} - \alpha_k$.
- Columns for crossed effects are computed by taking the horizontal direct product of columns corresponding to the constituent effects.

This coding corresponds to modeling without *over-parameterization*, using the same method as the CATMOD procedure in SAS/STAT software. This is different from the method used by the GLM procedure, which uses an over-parameterized model.

Orthogonal Coding

If you specify CODING=ORTH or CODING=ORTHCAN, the points are first coded as described in the previous section and then recoded so that $X_C'X_C = N_C \cdot I$, where X_C is the design matrix for the candidate points, N_C is the number of candidates, and I is the identity matrix. This is required in order for the D- and A-efficiency measures to make sense. For the option CODING=ORTHCAN, this recoding is accomplished by computing a square matrix R such that $X_C'X_C = R'R$ and then transforming each row vector $\mathbf{x}$ as

$$\mathbf{x} \quad \rightarrow \quad \mathbf{x}R^{-1}\sqrt{N_C}$$

If you specify CODING=ORTH, the recoding is done in a similar fashion, except that the matrix R is computed according to $X_C'X_C + X_A'X_A + X_I'X_I = R'R$, where X_A and X_I are the design matrices for the AUGMENT= and INITDESIGN= datasets, respectively (coded as described in the previous section.) Thus, these two orthogonal coding options only differ when there is an AUGMENT= or an INITDESIGN= data set (see pages 901–902); the option CODING=ORTH includes points from these data sets in computing the orthogonal coding, while the option CODING=ORTHCAN uses only the candidates themselves.

Example of Coding

For example, consider a main effect model with one continuous variable X and one three-level classification variable A. The results of the various coding options are shown in Table 31.5.

Table 31.5. Different Types of Design Coding

Original Data		Design Matrix With CODING=NONE				Design Matrix With CODING=STATIC				Design Matrix With CODING=ORTH			
X	**A**		**X**	**A1**	**A2**		**X**	**A1**	**A2**		**X**	**A1**	**A2**
1	1	1	1	1	0	1	–1	1	0	1	–1.464	0.598	–0.707
2	2	1	2	0	1	1	–0.6	0	1	1	–0.878	–0.478	1.414
3	3	1	3	–1	–1	1	–0.2	–1	–1	1	–0.293	–1.554	–0.707
4	1	1	4	1	0	1	0.2	1	0	1	0.293	1.554	–0.707
5	2	1	5	0	1	1	0.6	0	1	1	0.878	0.478	1.414
6	3	1	6	–1	–1	1	1	–1	–1	1	1.464	–0.598	–0.707

The first column in each design matrix is an all-ones vector corresponding to the intercept, the next column corresponds to the linear effect of X, and the last two columns correspond to the two degrees of freedom for the main effect of A.

General Recommendations

Coding does not affect the relative ordering of designs by D-efficiency, and the same is true for G-efficiency and the average standard error of prediction. This is easy to see for the latter two measures, which are based on the variance of prediction, since how accurately a point is predicted should not be affected by how the independent variables are coded. For D-optimality, note again that coding corresponds to multiplying the design matrix on the right by some non-singular transformation A, which changes the determinant of the information matrix as follows:

$$|X'X| \;\rightarrow\; |A'X'XA| \;=\; |A'A||X'X| \;=\; |A|^2|X'X|$$

Thus, recoding simply multiplies the D-criterion by a constant that is the same for all designs. Note, however, that A-optimality is *not* invariant to coding.

Orthogonal coding will usually be the right one; it is not the default because it depends on the candidate set. Note, however, that for the distance-based criteria, if the distance between two points should be computed in terms of the actual values of the model variables instead of centered and scaled values, then you should specify CODING=NONE or NOCODE. The NOCODE option is also usually appropriate when the NOINT option is specified.

Optimality Criteria

An optimality criterion is a single number that summarizes how good a design is, and it is maximized or minimized by an optimal design. This section discusses in detail the optimality criteria available in the OPTEX procedure.

Types of Criteria

Two general types of criteria are available: *information-based* criteria and *distance-based* criteria.

The information-based criteria that are directly available are D- and A-optimality; they are both related to the information matrix $X'X$ for the design. This matrix is

important because it is proportional to the inverse of the variance-covariance matrix for the least-squares estimates of the linear parameters of the model. Roughly, a good design should "minimize" the variance $(X'X)^{-1}$, which is the same as "maximizing" the information $X'X$. D- and A-efficiency are different ways of saying how large $(X'X)$ or $(X'X)^{-1}$ are.

For the distance-based criteria, the candidates are viewed as comprising a point cloud in p-dimensional Euclidean space, where p is the number of terms in the model. The goal is to choose a subset of this cloud that "covers" the whole cloud as uniformly as possible (in the case of U-optimality) or that is as broadly "spread" as possible (in the case of S-optimality). These ideas of coverage and spread are defined in detail on page 942. The distance-based criteria thus correspond to the intuitive idea of filling the candidate space as well as possible.

The rest of this section discusses different optimality criterion in detail.

D-optimality

D-optimality is based on the determinant of the information matrix for the design, which is the same as the reciprocal of the determinant of the variance-covariance matrix for the least-squares estimates of the linear parameters of the model.

$$|X'X| \;=\; 1/|(X'X)^{-1}|$$

The determinant is thus a general measure of the size of $(X'X)^{-1}$. D-optimality is the most common criterion for computer-generated optimal designs, which is why it is the default criterion for the OPTEX procedure.

The D-optimality criterion has the following characteristics:

- D-optimality is the most computationally efficient criterion to optimize for the low-rank update algorithms of the OPTEX procedure, since each update depends only on the variance of prediction for the current design; see "Useful Matrix Formulas" on page 943.
- $|X'X|$ is inversely proportional to the size of a $100(1-\alpha)\%$ confidence ellipsoid for the least-squares estimates of the linear parameters of the model.
- $|X'X|^{1/p}$ is equal to the geometric mean of the eigenvalues of $X'X$.
- The D-optimal design is invariant to non-singular recoding of the design matrix.

$$|X'X| \;\rightarrow\; |A'X'XA| \;=\; |A'A||X'X| \;=\; |A|^2|X'X|$$

A-optimality

A-optimality is based on the sum of the variances of the estimated parameters for the model, which is the same as the sum of the diagonal elements, or trace, of $(X'X)^{-1}$. Like the determinant, the A-optimality criterion is a general measure of the size of $(X'X)^{-1}$. A-optimality is less commonly used than D-optimality as a criterion for computer optimal design. This is partly because it is more computationally difficult to update; see "Useful Matrix Formulas" on page 943. Also, A-optimality is *not* invariant to non-singular recoding of the design matrix; different designs will be optimal with different codings.

G- and I-optimality

Both G-efficiency and the average prediction variance are well-known criteria for optimal design. Both are based on the variance of prediction of the candidate points, which is proportional to $\mathbf{x}'(X'X)^{-1}\mathbf{x}$. As this formula shows, these two criteria are also related to the information matrix $X'X$. Minimizing the average prediction variance has also been called *I-optimality*, the "I" denoting integration over the candidate space.

It is possible to apply the search techniques available in the OPTEX procedure to these two criteria, but this turns out to be a poor way to find G- and I-optimal designs. One reason for this is that there are no efficient low-rank update rules (see "Useful Matrix Formulas" on page 943), so that the searches can take a very long time. More seriously, for G-optimality such a search often does not converge on a design with good G-efficiency. G-efficiency is simply too "rough" a criterion to be optimized by the relatively short steps of the search algorithms available in the OPTEX procedure.

However, the OPTEX procedure does offer an approach for finding G-efficient designs. Begin by searching for designs according to the default D-optimality criterion. Then, from the various designs found on the different tries, you can save the one that has the best G-efficiency by specifying the NUMBER=GBEST option in the OUTPUT statement. Since D- and G-efficiency are highly correlated over the space of all designs, this method usually results in adequately G-efficient designs, especially when the number of tries is large. See the ITER= option on page 903 for details on specifying the number of tries.

To find I-optimal designs, note that if the design is orthogonally coded then I-optimality is equivalent to the A-optimality, since the sum of the prediction variances of all points $\mathbf{x}$ in the candidate space $\mathcal{C}$ is

$$\begin{aligned}
\sum_{\mathbf{x}\in\mathcal{C}} \mathbf{x}'(X'X)^{-1}\mathbf{x} &= \sum_{\mathbf{x}\in\mathcal{C}} \operatorname{trace}\left(\mathbf{x}'(X'X)^{-1}\mathbf{x}\right) \\
&= \operatorname{trace}\left((X'X)^{-1}\sum_{\mathbf{x}\in\mathcal{C}}\mathbf{x}\mathbf{x}'\right) \\
&= \operatorname{trace}\left((X'X)^{-1}X_C'X_C\right) \\
&= N_C \cdot \operatorname{trace}\left((X'X)^{-1}\right)
\end{aligned}$$

where N_C is the number of candidate points and X_C is the design matrix for the candidate points. Thus, you can use the option CODING=ORTH in the PROC OPTEX statement together with the option CRITERION=A in the GENERATE statement to search for I-optimal designs.

Note that both G- and I-optimality are invariant to non-singular recoding of the design matrix, since the coding does not affect how well a point is predicted.

Distance-based Criteria

The distance-based criteria are based on the distance $d(\mathbf{x}, \mathcal{A})$ from a point $\mathbf{x}$ in the p-dimensional Euclidean space $\mathcal{R}^p$ to a set $\mathcal{A} \subset \mathcal{R}^p$. This distance is defined as follows:

$$d(\mathbf{x}, \mathcal{A}) = \min_{\mathbf{y} \in \mathcal{A}} ||\mathbf{x} - \mathbf{y}||$$

where $||\mathbf{x} - \mathbf{y}||$ is the usual p-dimensional Euclidean distance,

$$||\mathbf{x} - \mathbf{y}|| = \sqrt{(x_1 - y_1)^2 + \ldots + (x_p - y_p)^2}$$

U-optimality seeks to minimize the sum of the distances from each candidate point to the design.

$$\sum_{\mathbf{x} \in \mathcal{C}} d(\mathbf{x}, \mathcal{D})$$

where $\mathcal{C}$ is the set of candidate points and $\mathcal{D}$ is the set of design points. You can visualize the U criterion by associating with any design point those candidates to which it is closest. Thus, the design defines a *clustering* of the candidate set, and indeed cluster analysis has been used in this context. Johnson, Moore, and Ylvisaker (1990) consider a similar measure of design efficiency, but over infinite rather than finite candidate spaces. Computationally, the U-optimality criterion can be *very* difficult to optimize, especially if the matrix of all pairwise distances between candidate points does not fit in memory. In this case, the OPTEX procedure recomputes each distance as needed. When searching for a U-optimal design, you should start with a small version of the problem to get an idea of the computing resources required.

S-optimality seeks to maximize the harmonic mean distance from each design point to all the other points in the design.

$$\frac{N_D}{\sum_{\mathbf{y} \in \mathcal{D}} 1/d(\mathbf{y}, \mathcal{D} - \mathbf{y})}$$

For an S-optimal design, the distances $d(\mathbf{y}, \mathcal{D} - \mathbf{y})$ are large, so the points are as spread out as possible. Since the S-optimality criterion depends only on the distances between design points, it is usually computationally easier to compute and optimize than the U-optimality criterion, which depends on the distances between all pairs of candidate points.

Memory and Run-Time Considerations

The OPTEX procedure provides a computationally intensive approach to designing an experiment, and therefore some finesse is called for to make the most efficient use of computer resources.

The OPTEX procedure must retain the entire set of candidate points in memory. This is necessary because all of the search algorithms access these points repeatedly. If this requires more memory than is available, consider using knowledge of the problem to

reduce the set of candidate points. For example, for first- or second-order models, it is usually adequate to restrict the candidates to just the center and the edges of the experimental region or perhaps an even smaller set; see the introductory examples on page 883 and page 884.

The distance-based criteria (CRITERION=U and CRITERION=S) also require repeated access to the distance between candidate points. The procedure will try to fit the matrix of these distances in memory; if it cannot, it will recompute them as needed, but this will cause the search to be dramatically slower.

The run time of each search algorithm depends primarily on N_D, the size of the target design and on N_C, the number of candidate points. For a given model, the run times of the sequential, exchange, and DETMAX algorithms are all roughly proportional to both N_D and N_C (that is, $O(N_D)+O(N_C)$). The run times for the two simultaneous switching algorithms (FEDOROV and M_FEDOROV) are roughly proportional to the product of N_D and N_C (that is, $O(N_C N_D)$). The constant of proportionality is larger when searching for A-optimal designs because the update formulas are more complicated (see "Search Methods," which follows).

For problems where either N_D or N_C is large, it is a good idea to make a few test runs with a faster algorithm and a small number of tries before attempting to use one of the slower and more reliable search algorithms. For most problems, the efficiency of a design found by a faster algorithm will be within one or two percent of that for the best possible design, and this is usually sufficient if it appears that searching with a slower algorithm is infeasible.

Search Methods

The search procedures available in the OPTEX procedure offer various compromises between speed and reliability in finding the optimum. In general, the longer an algorithm takes to arrive at an answer, the more efficient is the resulting design, although this is not invariably true. The right search procedure for any specific case depends on the size of the problem, the relative importance of using the best possible design as opposed to a very good one, and the computing resources available.

Useful Matrix Formulas

All of the search algorithms are based on adding candidate points to the growing design and deleting them from a design that is too big. If $V = (X'X)^{-1}$ is the inverse of the information matrix for the design at any stage, then the change in V that results from adding a new point to the design (which adds a new row $\mathbf{x}$ to the design matrix) is

$$V \rightarrow V - \frac{V\mathbf{x}\mathbf{x}'V}{1+\mathbf{x}'V\mathbf{x}}$$

and the change in V that results from deleting the point $\mathbf{y}$ from the design is

$$V \rightarrow V + \frac{V\mathbf{y}\mathbf{y}'V}{1-\mathbf{y}'V\mathbf{y}}$$

It follows, for example, that adding $\mathbf{x}$ multiplies the determinant of the information matrix by $1+\mathbf{x}'V\mathbf{x}$, and likewise deleting $\mathbf{y}$ multiplies the determinant by $1-\mathbf{y}'V\mathbf{y}$. For any point $\mathbf{z}$, the quantity $\mathbf{z}'V\mathbf{z}$ is proportional to the prediction variance at the point $\mathbf{z}$. Thus, the point $\mathbf{x}$ whose addition to the design maximizes the determinant of the information is the point whose prediction variance calculated from the present design is largest. The point whose deletion from the design costs the least in terms of the determinant is the point with the smallest prediction variance.

Similar rank-one update formulas can be derived for A-optimality, which is based on the trace of the inverse of the information matrix instead of its determinant. However, in this case there is no single quantity that can be examined for both adding and deleting a point. Instead, the trace that results from adding a point $\mathbf{x}$ depends on

$$\frac{\mathbf{x}'V^2\mathbf{x}}{1+\mathbf{x}'V\mathbf{x}}$$

and the trace that results from deleting a point $\mathbf{y}$ depends on

$$\frac{\mathbf{y}'V^2\mathbf{y}}{1-\mathbf{y}'V\mathbf{y}}$$

This complication makes A-optimal designs harder to search for than D-optimal ones.

There are no useful rank-one update formulas for the distance-based design criteria.

Sequential Search Algorithm

The simplest and fastest algorithm is the sequential search due to Dykstra (1971), which starts with an empty design and adds successive candidate points so that the chosen criterion is optimized at each step. You can use the sequential procedure as a first step in finding a design

- to judge the size of the problem in terms of time and space requirements
- to determine the number of design points needed to estimate the parameters of the model

The sequential algorithm requires no initial design; in fact, it can be used to provide an initial design for the other search procedures (see the INITDESIGN= option on page 902). If you specify a data set for an initial design for this search procedure, no search will be made; in this way, the OPTEX procedure can be used to evaluate an existing design.

Since the sequential search method involves no randomness, it requires only one try to find a design. The sequential procedure is by far the fastest of any of the search methods, but in difficult design situations it is also the least reliable in finding a globally optimal design. Also, the fact that it always finds the same design (due to the lack of randomness mentioned previously) makes it inappropriate when you want to find a design that represents a compromise between several optimality criteria.

Exchange Algorithm

The next fastest algorithm is the simple exchange method of Mitchell and Miller (1970). This technique tries to improve an initial design by adding a candidate point and then deleting one of the design points, stopping when the chosen criterion ceases to improve. This method is relatively fast (though typically much slower than the sequential search) and fairly reliable. METHOD=EXCHANGE is the default.

Johnson and Nachtsheim (1983) introduce a generalization of both the simple exchange algorithm and the modified Fedorov search algorithm of Cook and Nachtsheim (1980), which is described later in this list. In the modified Fedorov algorithm, each of the points in the current design is considered for exchange with a candidate point; in the generalized version, only the k design points with smallest variance in the current design are considered for exchange. You can specify k-exchange as the search procedure for OPTEX by giving a value for k in parentheses after METHOD=EXCHANGE. When $k = N_D$, the size of the design, k-exchange is equivalent to the modified Fedorov algorithm; when $k = 1$, it is equivalent to the simple exchange algorithm. Cook and Nachtsheim (1980) indicate that $k < N_D/4$ is typically sufficient.

DETMAX Algorithm

The DETMAX algorithm of Mitchell (1974a) is the best known and most widely used optimal design search algorithm. It generalizes the simple exchange method. Instead of requiring that each addition of a point be followed directly by a deletion, the algorithm provides for *excursions* in which the size of the design may vary between $N_D + k$ and $N_D - k$. Here $N_D + k$ is the specified size of the design and k is the maximum allowed size for an excursion. By default k is 4, but you can change this (see the METHOD=DETMAX(*level*) option on page 903). For the precise stopping rules for each excursion and for the entire search, refer to Mitchell (1974a).

Fedorov and Modified Fedorov Algorithms

The three algorithms discussed so far add and delete points one at a time. By contrast, the Fedorov and modified Fedorov algorithms are based on simultaneous switching, adding and deleting points simultaneously. These two algorithms usually find a better design than the others, but because each step involves a search over all possible pairs of candidate and design points, they generally run much slower.

At each step, the Fedorov algorithm (Fedorov 1972) seeks the pair $(\mathbf{x}, \mathbf{y})$ of one candidate point and one design point that optimizes the change $\Delta(\mathbf{x}, \mathbf{y})$ in the optimality criterion, and then switches $\mathbf{x}$ for $\mathbf{y}$ in the design. Thus, after computing $\Delta(\mathbf{x}, \mathbf{y})$ for all possible pairs of candidate and design points, the Fedorov algorithm performs only one switch.

The modified Fedorov algorithm of Cook and Nachtsheim (1980) computes the same number of Δ's on each step but switches each point $\mathbf{y}$ in the design with the candidate point $\mathbf{x}$ that maximizes Δ(**x,y**). This procedure is generally as reliable as the simple Fedorov algorithm in finding the optimal design, but it can be up to twice as fast.

Optimal Blocking

Building on the work of Harville (1974), Cook and Nachtsheim (1989) give an algorithm for finding D-optimal designs in the presence of fixed block effects. In this case, the design for the original candidate points is called the *treatment design*; the information matrix for the treatment design has the form $X'AX$ for a certain symmetric, nonnegative-definite matrix A that depends on the blocks. The algorithm is based on two kinds of low-rank changes to the treatment design matrix X: *exchanging* a point in the design with a potential treatment point, and *interchanging* two points in the design. Cook and Nachtsheim (1989) give formulas for computing the resulting change in $X'AX$ and $|X'AX|$. These update formulas can be generalized to apply whenever the information matrix for the treatment design has the form $X'AX$, not just when A is derived from fixed blocks. This is the basis for the optimal blocking algorithm in the OPTEX procedure.

Notice that you can combine several options to use the OPTEX procedure to *evaluate* a design with respect to the fixed covariates. Assume the design you want to evaluate is in a data set named EDESIGN. Then first specify

```
generate initdesign=edesign method=sequential;
```

This makes the data set EDESIGN the treatment design. Then specify the following BLOCKS statement options:

```
blocks {block-specification} init=chain iter=0;
```

The INIT=CHAIN option ensures that the starting ordering for the treatment points is the same as in the EDESIGN data set, and the ITER=0 specification causes the procedure simply to output the efficiencies for the initial design, without trying to optimize it.

Search Strategies

General Recommendations

As with all combinatorial optimization problems, finding efficient experimental designs can be difficult. For this reason, the OPTEX procedure provides a variety of ways to customize the search.

Although default settings make the procedure simple to use "as is," you can usually improve the search using knowledge of the specific design problem. For example, if the default algorithm (EXCHANGE) runs quickly but it is not clear whether it finds the best design, you can try a slower but more reliable search method or use more iterations than the default number of 10.

Set of Candidate Points

The choice of candidate points can profoundly affect both the speed with which the search converges at a local optimum and the likelihood that this local optimum is indeed the global optimum. Up to a point, the more candidate points there are, the

better the resulting optimum design will be but the longer it will take to find. Any prior knowledge that can be brought to bear on the choice of candidates will almost certainly improve the search. For example, for first- or second-order models it is usually adequate to restrict the candidates to just the center and the edges of the experimental region, or perhaps even less; refer to Snee (1985), and see the introductory examples on page 883 and page 884.

Initial Design

The reliability of the search algorithms in finding the optimal design can be quite sensitive to the choice of initial design. The default method of initialization for each search procedure should achieve good results for a wide variety of situations (see the INITDESIGN= option on page 902). However, in certain situations it is better to override the defaults. For example, if there are many local optima and you want to find the exact global optimum, it will probably be best to start each try with a completely random design (INITDESIGN=RANDOM). On the other hand, prior knowledge may provide a specific initial design, which can be placed in a SAS data set and specified with the INITDESIGN= option.

Output

By default, the OPTEX procedure lists the following information for each attempt to find the optimum design:

- the D-efficiency of the design
- the A-efficiency of the design
- the G-efficiency of the design
- the square root of the average variance for prediction over the candidate points

If you specify a BLOCKS statement, then the covariate-adjusted D- and A-efficiencies are also listed.

See "Design Efficiency Measures" on page 936 for details on the efficiencies. The OPTEX procedure orders the designs first by the optimality criteria with which they were generated and then by optimality with respect to the other three preceding measures.

If you use the NOCODE option, the OPTEX procedure lists

- $\log|\mathbf{X}'\mathbf{X}|$
- $\mathrm{trace}(X'X)^{-1}$
- the G-efficiency of the design
- the square root of the average variance for prediction over the candidate points

If you specify one of the distance-based optimality criteria (CRITERION=U or CRITERION=S), then, instead of the preceding efficiencies, alternative measures of coverage and spread are listed. For U-optimality these measures are

- the average distance from each candidate to the nearest design point (this is the U criterion)
- the average harmonic mean distance from each candidate to the design

For S-optimality, the following alternative measures of spread are listed:

- the harmonic mean distance from each design point to the nearest other design point (this is the S criterion)
- the average distance from each design point to the nearest other design point

In addition, the OPTEX procedure can create an output data set, as described in "OUTPUT Statement" on page 905 and in "Output Data Sets" on page 933.

ODS Tables

The following table summarizes the ODS tables that you can request with the PROC OPTTEX statement.

Table 31.6. ODS Tables Produced in PROC OPTEX

ODS Table Name	Description	Statement	Option
ClassLevels	Classification variable levels	CLASS	default
FactorRanges	Continuous variable ranges	default	default
BlockDesignEfficiencies	Block design efficiency criteria	BLOCK	default
Efficiencies	Efficiency criteria for all designs	GENERATE	default
Criteria	Efficiency criteria for a single design	EXAMINE	default
Points	Design points	EXAMINE	POINTS
Information	Information matrix XPX	EXAMINE	INFORMATION
Variance	Inverse information matrix inv(XPX)	EXAMINE	VARIANCE
Status	Optimization status	PROC	STATUS
Distances	Distance criteria for all designs	GENERATE	CRITERION=U or S

References

Atkinson, A.C. and Donev, A.N. (1992), *Optimum Experimental Designs*, New York: Oxford University Press.

Cochran, W.G. and Cox, G.M. (1957), *Experimental Designs, Second Edition,* New York: John Wiley & Sons, Inc.

Cook, R.D. and Nachtsheim, C.J. (1980), "A Comparison of Algorithms for Constructing Exact D-optimal Designs," *Technometrics*, 22, 315–324.

Cook, R.D. and Nachtsheim, C.J. (1989), "Computer-Aided Blocking of Factorial and Response-Surface Designs," *Technometrics*, 31, 339–346.

DuMouchel, W. and Jones, B. (1994), "A Simple Bayesian Modification of D-Optimal Designs to Reduce Dependence on an Assumed Model," *Technometrics*, 36, 37–47.

Dykstra, O., Jr. (1971), "The Augmentation of Experimental Data to Maximize $|X'X|$," *Technometrics*, 13, 682–688.

Fedorov, V.V. (1972), *Theory of Optimal Experiments*, translated and edited by W.J. Studden and E.M. Klimko, New York: Academic Press.

Galil, Z. and Kiefer, J. (1980), "Time- and Space-saving Computer Methods, Related to Mitchell's DETMAX, for Finding D-Optimum Designs," *Technometrics*, 22, 301–313.

Harville, D. (1974), "Nearly Optimal Allocation of Experimental Units Using Observed Covariate Values," *Technometrics*, 16, 589–599.

Johnson, M.E., Moore, L.M., and Ylvisaker, D. (1990), "Minimax and Maximin Distance Designs," *Journal of Statistical Planning and Inference*, 26, 131–148.

Johnson, M.E. and Nachtsheim, C.J (1983), "Some Guidelines for Constructing Exact D-optimal Designs on Convex Design Spaces," *Technometrics*, 25, 271–277.

Mitchell, T.J. (1974a), "An Algorithm for the Construction of D-optimal Experimental Designs," *Technometrics*, 16, 203–210.

Mitchell, T.J. (1974b), "Computer Construction of 'D-optimal' First-Order Designs," *Technometrics*, 16, 211–220.

Mitchell, T.J. and Miller, F.L., Jr. (1970), "Use of Design Repair to Construct Designs for Special Linear Models," *Math. Div. Ann. Progr. Rept. (ORNL-4661)*, 130–131, Oak Ridge, TN: Oak Ridge National Laboratory.

SAS Institute Inc. (1999), *SAS/STAT User's Guide, Version 8*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1999), *Getting Started with the ADX Interface for Design of Experiments*, Cary, NC: SAS Institute Inc.

Searle, S.R. (1971), *Linear Models*, New York: John Wiley & Sons, Inc.

Snee, R.D. (1985), "Computer-aided Design of Experiments–Some Practical Experiences," *Journal of Quality Technology*, 17, 222–236.

Vance, L.C. (1986), "Computer Construction of Experimental Designs," General Motors Research Report GMR-5411, General Motors Laboratories, Warren, Michigan.

Part 8
The PARETO Procedure

Contents

Introduction 953

Chapter 32. PROC PARETO Statement 955

Chapter 33. VBAR Statement 961

Chapter 34. HBAR Statement 997

Chapter 35. INSET Statement 1031

Chapter 36. Details and Examples 1047

References 1077

Introduction

The PARETO procedure creates Pareto charts, which display the relative frequency of quality-related problems in a process or operation. The frequencies are represented by bars that are ordered in decreasing magnitude. Thus, a Pareto chart can be used to decide which subset of problems should be solved first or which problem areas deserve the most attention.

Pareto charts provide a tool for visualizing the Pareto principle,* which states that a small subset of problems tend to occur much more frequently than the remaining problems. In Japanese industry, the Pareto chart is one of the "seven basic QC tools" heavily used by workers and engineers. Ishikawa (1976) discusses how to construct and interpret a Pareto diagram. Examples of Pareto diagrams are also given by Kume (1985) and Wadsworth and others (1986).

You can use the PARETO procedure to

- construct Pareto charts from unsorted raw data (for instance, a set of quality problems that have not been classified into categories) or from a set of distinct categories and corresponding frequencies
- construct Pareto charts based on the percentage of occurrence of each problem, the frequency (number of occurrences), or a weighted frequency (such as frequency weighted by the cost of each problem)
- add a curve indicating the cumulative percentage across categories
- construct side-by-side Pareto charts or stacked Pareto charts
- construct *comparative Pareto charts* that enable you to compare the Pareto frequencies across levels of one or two classification variables. For example, you can compare the frequencies of problems encountered with three different machines for five consecutive days.
- highlight the "vital few" and the "useful many"† categories by using different colors for bars corresponding to the n most frequently occurring categories or the m least frequently occurring categories.
- create charts with bars oriented vertically or horizontally
- highlight special categories by using different colors for specific bars
- create charts using either a high-resolution graphics device or a line printer
- annotate charts created on graphics devices
- inset summary statistics in charts created on graphics devices

*Both the chart and the principle are named after Vilfredo Pareto (1848-1923), an Italian economist and sociologist. His first work, *Cours d'Économie Politique* (1895-1897), applied what is now termed the *Pareto distribution* to the study of income size.

†Juran originally referred to these categories as the "trivial many"; however, because all problems merit attention, the term "useful many" is preferable. Refer to Burr (1990).

- save charts created on graphics devices in a graphics catalog for subsequent replay
- display sample sizes on Pareto charts
- display frequencies above the bars
- define characters used for features on plots produced on line printers
- save information associated with the categories (such as the frequencies) in an output data set
- restrict the number of categories displayed to the n most frequently occurring categories

Chapter 32
PROC PARETO Statement

Chapter Contents

OVERVIEW 957

SYNTAX 958
Summary of Options 958
Dictionary of Options 959

Chapter 32
PROC PARETO Statement

Overview

The PROC PARETO statement starts the PARETO procedure, and it optionally identifies various data sets and requests line printer charts.

To create a Pareto chart, you specify a chart statement (VBAR or HBAR) after the PROC PARETO statement. The chart statement specifies the type of Pareto chart you want to create and the variables in the input data set that you want to analyze. A VBAR statement produces a chart with vertical bars; an HBAR statement produces a chart with horizontal bars. For example, the following statements request a horizontal Pareto chart with blue bars on a green background:

```
proc pareto data=failures;
   hbar reason1 / cbars  = blue
                  cframe = green;
run;
```

The DATA= option specifies the input data set (FAILURES) containing the variable (REASON1) to be analyzed.

A Pareto chart has three axes:

- the *category axis*
- the *frequency axis* and
- the *cumulative percent axis*.

On a vertical bar chart the category axis is displayed horizontally at the bottom of the chart. The frequency axis (also called the *primary vertical axis*) is on the left. The cumulative percent axis (or *secondary vertical axis*) is displayed on the right.

On a horizontal bar chart the category axis is displayed vertically on the left side of the chart. The frequency axis (*primary horizontal axis*) is displayed at the top of the chart. The cumulative percent axis (*secondary horizontal axis*) is at the bottom.

Syntax

The syntax for the PARETO procedure is as follows:

PROC PARETO < *options* >**;**

VBAR (*variable-list*) < / *options* > ;
HBAR (*variable-list*) < / *options* > ;
INSET (*keyword-list*) < / *options* > ;
< **BY** *variables* ; >

You must specify the PROC PARETO statement and at least one VBAR or HBAR statement (also referred to as chart statements). All other statements, such as INSET, TITLE and BY statements, are optional. If you specify two or more *variables* in a chart statement, they must be enclosed in parentheses. You can use multiple chart statements with one PROC PARETO statement. An INSET statement must immediately follow a chart statement. It produces an inset displaying information on the chart created by the chart statement it follows. For details on the VBAR, HBAR or INSET statement, read the chapter on that statement.

Summary of Options

The following table lists the PROC PARETO statement options.

For complete descriptions, see "Dictionary of Options" on page 959.

Table 32.1. PROC PARETO Statement Options

ANNOTATE=*SAS-data-set*	specifies annotate data set for frequency axis
ANNOTATE2=*SAS-data-set*	specifies annotate data set for cumulative percent axis
DATA=*SAS-data-set*	specifies input data set
FORMCHAR=*'string'*	specifies form character list to enhance line printer charts
GOUT=*graphics catalog*	specifies graphics catalog for saving graphics output
LINEPRINTER	creates charts for a line printer device

Dictionary of Options

You can specify the following options in the PROC PARETO statement. The marginal notes, *Graphics* and *Line Printer*, identify options that apply only to graphics devices and line printers, respectively.

ANNOTATE=*SAS-data-set*
ANNO=*SAS-data-set*

specifies an input data set that contains annotate variables as described in *SAS/GRAPH Software: Reference*. You can use this data set to customize charts with features such as labels explaining critical categories. The ANNOTATE= data set is associated with the frequency axis. If the annotation is based on data coordinates, you must use the same units as the frequency axis. Features provided in this data set are added to every chart produced in the current run of the procedure. *Graphics*

ANNOTATE2=*SAS-data-set*
ANNO2=*SAS-data-set*

specifies an input data set that contains annotate variables as described in *SAS/GRAPH Software: Reference*. You can use this data set to customize charts with features such as labels explaining critical categories. The ANNOTATE2= data set is associated with the cumulative percent axis. If the annotation is based on data coordinates, you must use the same units as the cumulative percent axis. Features provided in this data set are added to every chart produced in the current run of the procedure. *Graphics*

DATA=*SAS-data-set*

specifies an input data set that contains the *process variables* and related variables. If you do not specify a DATA= data set, the procedure uses the most recently created data set.

FORMCHAR=*'string'*

specifies a form character list that enhances the appearance of line printer charts with corner characters and other special characters. *Line Printer*

If your device supports the ASCII symbol set (1 or 2), use the following list:

```
formchar = 'B3,C4,DA,C2,BF,C3,C5,B4,C0,C1,D9'X
```

The FORMCHAR= option overrides (but does not alter) the FORMCHAR= option that is specified with an OPTIONS statement such as

```
options formchar = 'B3,C4,DA,C2,BF,C3,C5,B4,C0,C1,D9'X ;
```

You can place the OPTIONS statement at the top of your SAS program or in an AUTOEXEC.SAS file.

GOUT=*graphics-catalog*

specifies the graphics catalog in which to save graphics output. *Graphics*

LINEPRINTER

requests that line printer charts be produced. By default, the procedure creates charts for a graphics device. The HBAR statement does not produce line printer output, so you cannot use an HBAR statement when the LINEPRINTER option is specified. *Line Printer*

Chapter 33
VBAR Statement

Chapter Contents

OVERVIEW 963

GETTING STARTED 963
- Creating a Pareto Chart from Raw Data 963
- Creating a Pareto Chart Using Frequency Data 966
- Restricting the Number of Pareto Categories 968

SYNTAX 970
- Summary of Options 971
- Dictionary of Options 976

Chapter 33
VBAR Statement

Overview

The VBAR statement creates a Pareto chart with vertical bars representing the frequencies of problems in a process or operation. A vertical Pareto chart has one horizontal axis on which the Pareto categories are listed. The primary vertical axis appears on the left side of the chart and is used to read the lengths of the bars on the chart. The secondary vertical axis is on the right of the chart and is used to read the cumulative percent curve.

Getting Started

The examples in this section illustrate basic features of the VBAR statement. Complete syntax for the VBAR statement is presented in the "Syntax" section on page 970.

Creating a Pareto Chart from Raw Data

In the fabrication of integrated circuits, common causes of failures include improper doping, corrosion, surface contamination, silicon defects, metallization, and oxide defects. The causes of 31 failures were recorded in a SAS data set called FAILURE1.

See PARETO4 in the SAS/QC Sample Library

```
data failure1;
   length cause $ 16 ;
   label  cause = 'Cause of Failure' ;
   input  cause $ 3-18 ;
   datalines;
  Corrosion
  Oxide Defect
  Contamination
  Oxide Defect
  Oxide Defect
  Miscellaneous
  Oxide Defect
  Contamination
  Metallization
  Oxide Defect
  Contamination
  Contamination
  Oxide Defect
  Contamination
  Contamination
  Contamination
  Corrosion
  Silicon Defect
```

```
  Miscellaneous
  Contamination
  Contamination
  Contamination
  Miscellaneous
  Contamination
  Contamination
  Doping
  Oxide Defect
  Oxide Defect
  Metallization
  Contamination
  Contamination
;
run;
```

Each of the 31 observations corresponds to a different circuit, and the value of CAUSE provides the cause for the failure. These are raw data in the sense that there is more than one observation with the same value of CAUSE, and the observations are not sorted by CAUSE. The following statements produce a basic Pareto chart for the failures:

```
proc pareto data=failure1;
   vbar cause;
run;
```

The PARETO procedure is invoked with the first statement, referred to as the PROC statement. You specify the process variable to be analyzed in the VBAR statement.

The Pareto chart is shown in Figure 33.1.

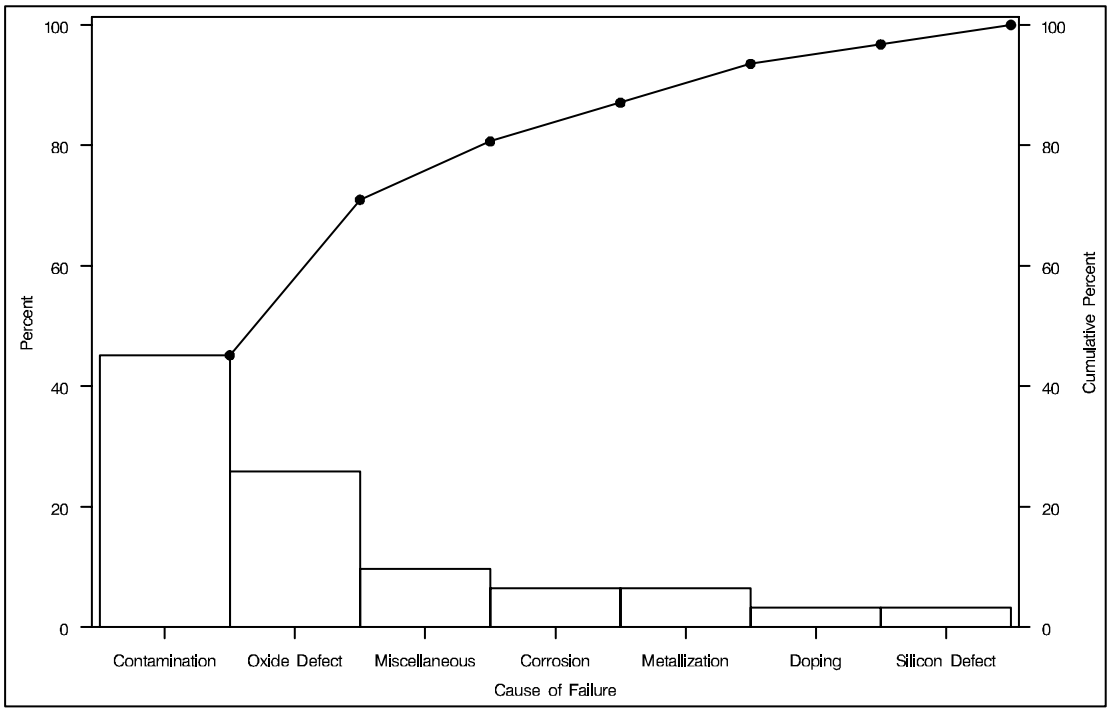


Figure 33.1. Pareto Chart for IC Failures in the Data Set FAILURE1

The procedure has classified the values of CAUSE into seven distinct categories (levels). The bars represent the percent of failures in each category, and they are arranged in decreasing order. Thus, the most frequently occurring category is *Contamination*, which accounts for 45% of the failures. The Pareto curve indicates the cumulative percent of failures from left to right; for example, *Contamination* and *Oxide* together account for 71% of the failures.

If there is sufficient space, the procedure labels the bars along the horizontal axis as in Figure 33.1. Otherwise, as in Figure 33.2, the procedure numbers the bars from left to right and adds a legend identifying the categories.

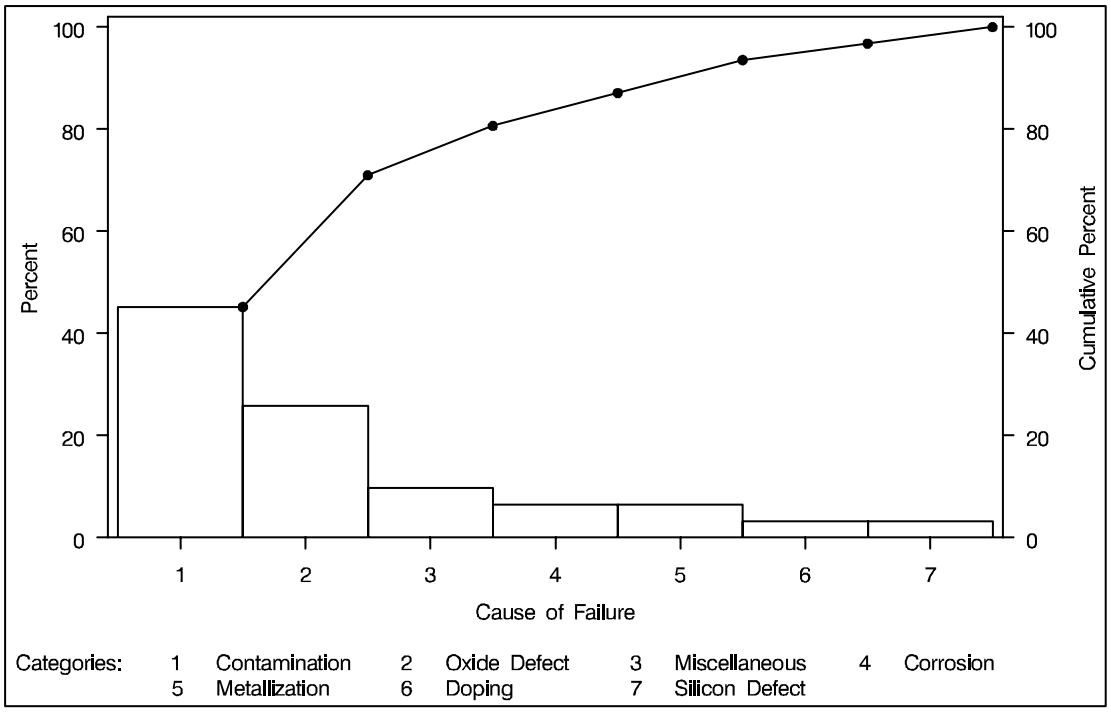


Figure 33.2. Pareto Chart with Category Legend

A category legend is likely to be introduced when

- the number of categories is large
- the category labels are lengthy (as in this example). Category labels can be up to 32 characters.
- a large text height is used. You can specify the height with the HEIGHT= option in the VBAR statement or with the HTEXT= option in a GOPTIONS statement (not shown here).

Creating a Pareto Chart Using Frequency Data

See PARETO5 in the SAS/QC Sample Library

In some situations, a count (frequency) is available for each category, or you can compress a large data set by creating a frequency variable for the categories before applying the PARETO procedure.

For example, you can use the FREQ procedure to obtain the compressed data set FAILURE2 from the data set FAILURE1.

```
proc freq data=failure1;
   tables cause / noprint out=failure2;
run;


proc print data=failure2;
run;
```

A listing of FAILURE2 is shown in Figure 33.3.

Obs	cause	COUNT	PERCENT
1	Contamination	14	45.1613
2	Corrosion	2	6.4516
3	Doping	1	3.2258
4	Metallization	2	6.4516
5	Miscellaneous	3	9.6774
6	Oxide Defect	8	25.8065
7	Silicon Defect	1	3.2258

Figure 33.3. The Data Set FAILURE2 Created Using PROC FREQ

The following statements produce a Pareto chart for the data in FAILURE2:

```
title 'Analysis of IC Failures';
proc pareto data=failure2;
   vbar cause / freq       = count
                scale      = count
                interbar   = 1.0
                last       = 'Miscellaneous'
                nlegend    = 'Total Circuits'
                cframenleg = empty;
run;
```

The chart is displayed in Figure 33.4.

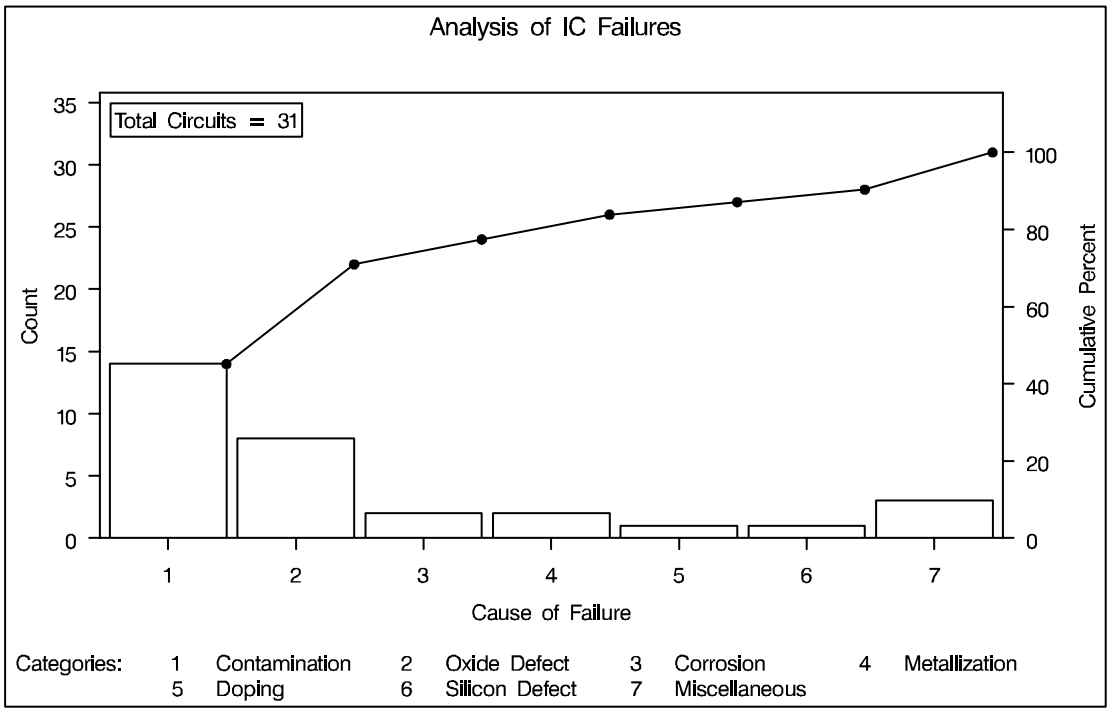


Figure 33.4. Pareto Chart with Frequency Scale

A slash (/) is used to separate the process variable CAUSE from the options specified in the VBAR statement. The frequency variable COUNT is specified with the FREQ= option. Specifying the keyword COUNT with the SCALE= option requests a frequency scale for the vertical axis.

The INTERBAR= option inserts a small space between the bars, and specifying LAST='Miscellaneous' causes the category *Miscellaneous* to be displayed last regardless of its frequency. The NLEGEND= option adds a sample size legend labeled *Total Circuits*, and the CFRAMENLEG= option frames the legend. The SYMBOL statement marks points on the curve with dots.

In the preceding statements, adding the keyword LINEPRINTER to the PROC PARETO statement requests a line printer version of the chart, which is displayed in Figure 33.5. *

*In Release 6.12 and previous releases of SAS/QC software, the keyword GRAPHICS was required in the PROC PARETO statement to specify that the chart be created with a graphics device. In Version 7 and later releases, you can specify the LINEPRINTER option to request line printer plots.

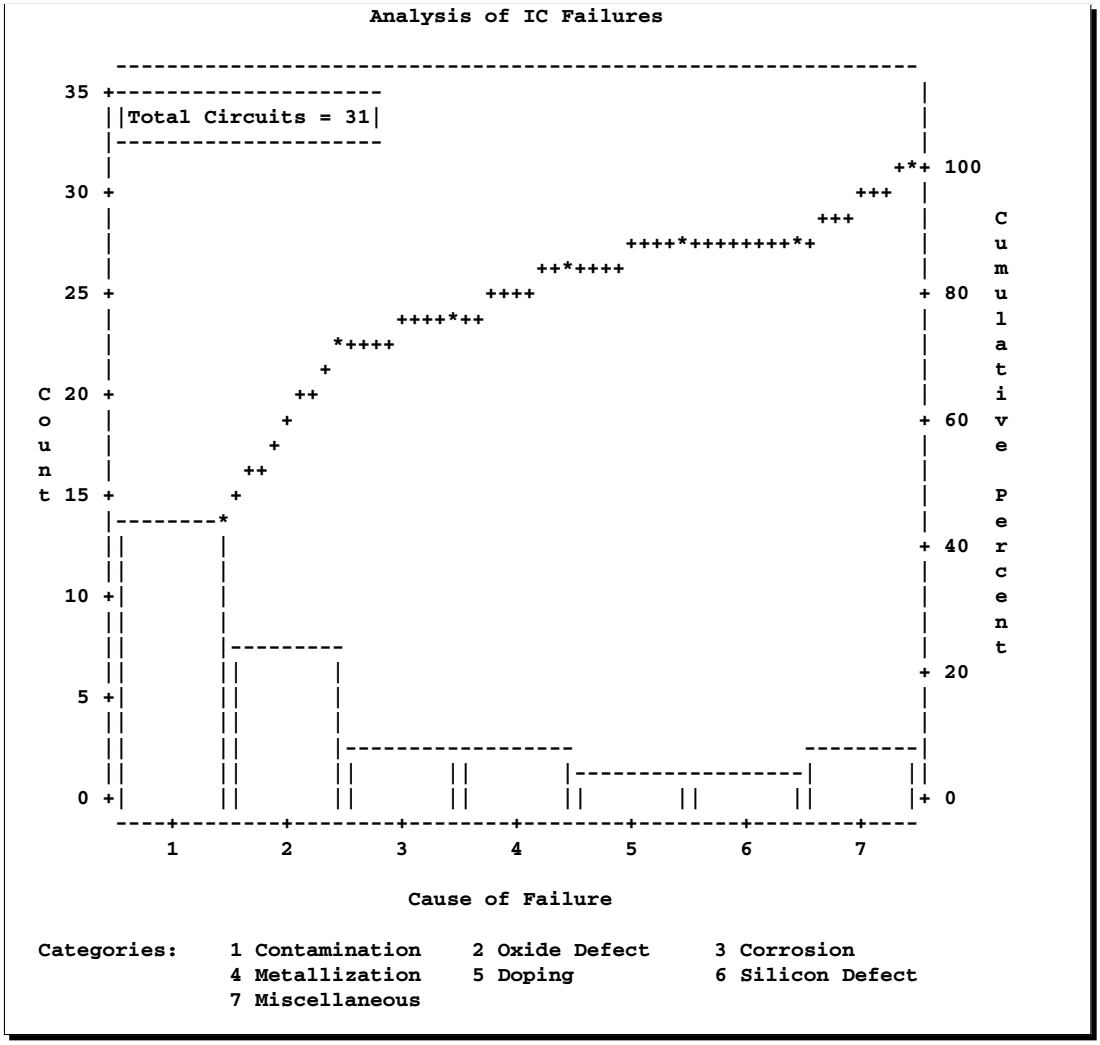


Figure 33.5. Pareto Chart for IC Failures in the Data Set FAILURE2

There are two sets of tied categories in this example; *Corrosion* and *Metallization* each occur twice, and *Doping* and *Silicon Defect* each occur once. The procedure displays tied categories in alphabetical order of their formatted values. Thus, *Corrosion* appears before *Metallization*, and *Doping* appears before *Silicon Defect* in Figure 33.4 and Figure 33.5. This is simply a convention, and no practical significance should be attached to the order in which tied categories are arranged.

Restricting the Number of Pareto Categories

See PARETO2 in the SAS/QC Sample Library

Unlike the previous examples, some applications involve too many categories to display on a chart. The solution presented here is to create a restricted Pareto chart that displays only the most frequently occurring categories.

The following statements create a Pareto chart for the five most frequently occurring levels of CAUSE in the data set FAILURE2, which is listed in Figure 33.3:

```
title 'IC Manufacturing Problems';
proc pareto data=failure2;
   vbar cause / freq     = count
                scale    = count
                maxncat  = 5;
run;
```

The MAXNCAT= option specifies the number of categories to be displayed. The chart, shown in Figure 33.6, does not display the categories *Doping* and *Silicon Defect*.

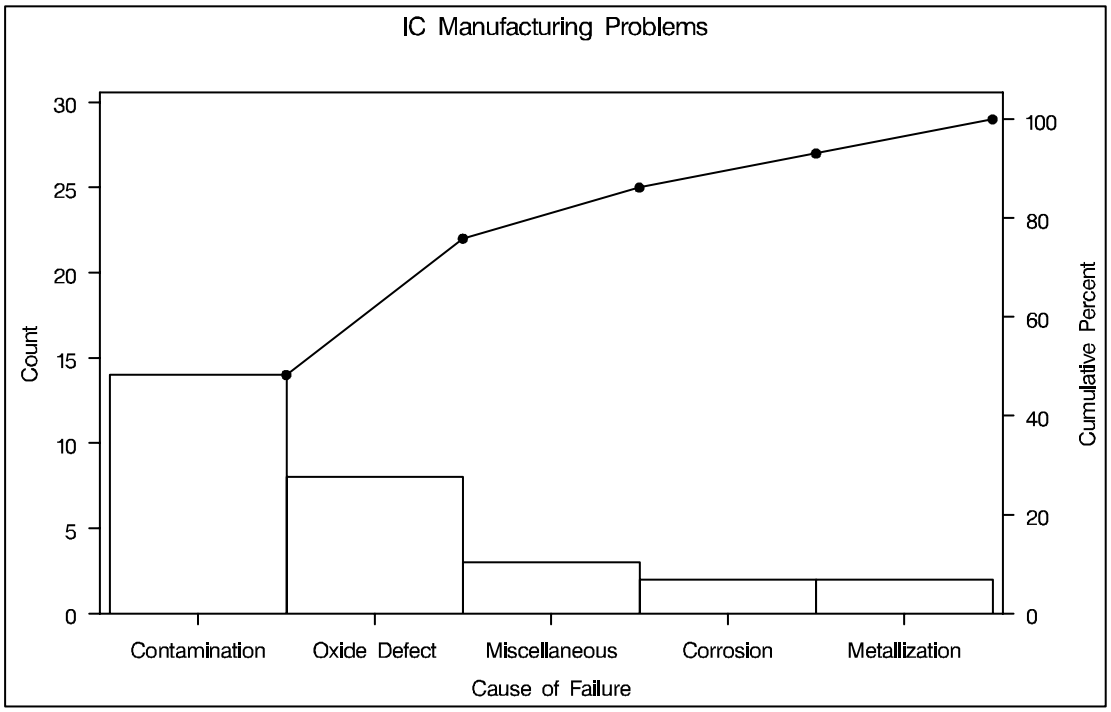


Figure 33.6. Restricted Pareto Chart

You can also display the most frequently occurring categories and merge the remaining categories into a single *other* category that is represented by a bar. You can specify the name for the new category with the OTHER= option. If, in addition, you specify the name with the LAST= option, the category is positioned at the far right of the chart. The following statements illustrate both options:

```
title 'IC Manufacturing Problems';
proc pareto data=failure2;
   vbar cause / freq       = count
                scale      = count
                maxncat    = 5
                other      = 'Others'
                last       = 'Others';
run;
```

The chart is shown in Figure 33.7.

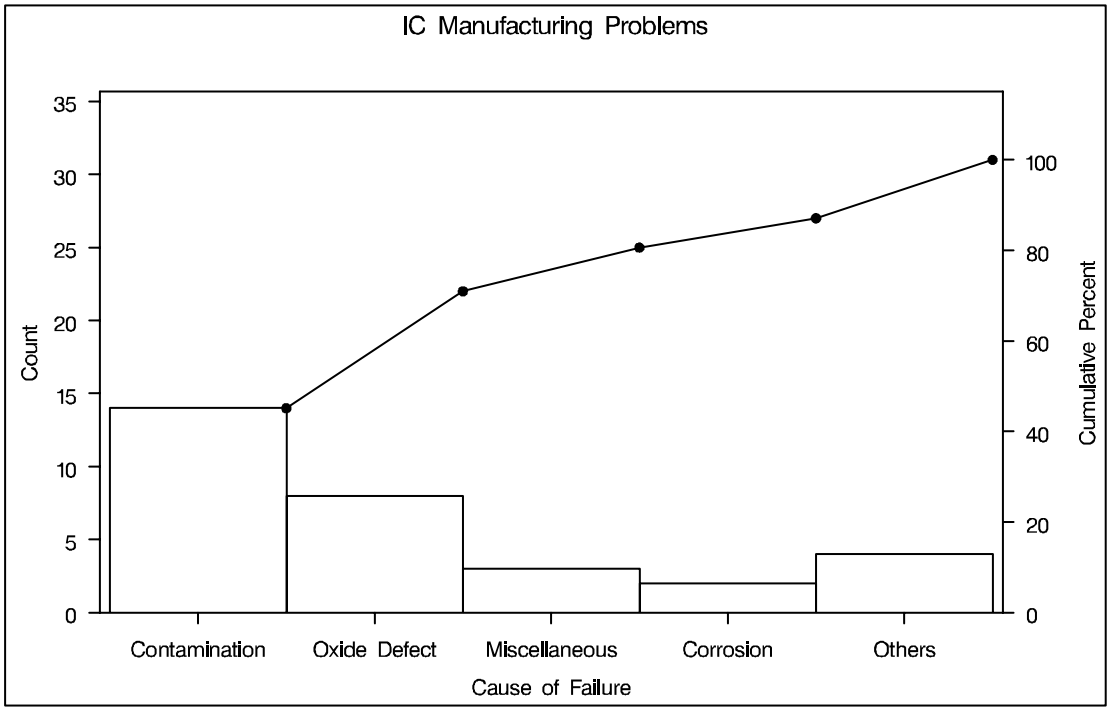


Figure 33.7. Restricted Pareto Chart with *Other* Category

The number of categories displayed is five, which is the number specified with the MAXNCAT= option. The first four categories are the four most frequently occurring problems in FAILURE2, and the fifth category merges the remaining problems.

Note that *Corrosion* and *Metallization* both have a frequency of two. When the MAXNCAT= option is applied to categories with tied frequencies, the procedure breaks the tie by using the order of the formatted values. Thus *Corrosion* is displayed, whereas *Metallization* is merged into the *Other* category. The MAXNCAT= and related options are described in "Restricted Pareto Charts" on page 1050.

Syntax

The syntax for the VBAR statement is as follows:

VBAR (*variable-list***) < / *options* > ;**

You can use any number of VBAR statements in the PARETO procedure. If you specify two or more *variables* in the VBAR statement, they must be enclosed in parentheses. The components of the VBAR statement are described as follows.

options
specify the layout and features of the chart, and they are listed after a slash (/) that follows the variables to be analyzed.

The "Summary of Options" section, which follows, provides summary tables of options organized by function. The "Dictionary of Options" on page 976 describes the *options* in detail.

variable-list

specifies the process variables to be analyzed. A chart is created for each *variable*, and the values of each *variable* determine the Pareto categories for that chart. A list of two or more *variables* must be enclosed in parentheses.

The *variables* can be numeric or character, and the maximum length of a character *variable* is 32. Formatted values are used to determine the categories and are displayed in labels and legends. The maximum format length is 32.

Summary of Options

The following tables list the VBAR statement options by function. For complete descriptions, see "Dictionary of Options" on page 976.

Table 33.1. Options for the Cumulative Percent Curve

ANCHOR=*keyword*	specifies corner of leftmost bar to which curve is anchored
CCONNECT=*color*	specifies color for curve
CMPCTLABEL	labels curve points with their values
CONNECTCHAR= *'character'*	specifies plot character for curve segments
NOCURVE	suppresses curve
NOVLABEL2	suppresses secondary vertical axis label
NOVTICK2	suppresses secondary vertical axis tick marks and tick mark labels
SYMBOLCHAR= *'character'*	specifies plot character for points on curve

Table 33.2. Data Processing Options

CFRAMENLEG=*color*	frames the NLEGEND legend and fills the frame with the specified color
FREQ=*variable*	specifies frequency variable
MISSING	specifies that missing values of the process variable be treated as a Pareto category
MISSING1	specifies that missing values of the first CLASS= variable be analyzed as a level
MISSING2	specifies that missing values of the second CLASS= variable be analyzed as a level
NLEGEND	requests sample size legend
NLEGEND='*label*' | (*variable*)	requests sample size legend with specified label
OUT=*SAS-data-set*	creates output data set that saves information displayed in the Pareto chart
WEIGHT=*variable-list*	specifies weight variables used to weight frequencies

Table 33.3. Options for Restricting the Number of Categories

COTHER=*color*	specifies color for OTHER= bar
LOTHER='*label*'	specifies label for OTHER= bar
MAXCMPCT=*percent*	displays only the categories with cumulative percentage less than the *percent* specified
MAXNCAT=n	displays only the categories with the n highest values
MINPCT=*percent*	displays only the categories with percents greater than the *percent* specified
OTHER='*category*'	merges all categories not displayed
OTHERCVAL='*value*'	specifies an OUT= data set character variable value for the OTHER= category
OTHERNVAL=*value*	specifies an OUT= data set numeric variable value for the OTHER= category
POTHER=*pattern*	specifies pattern for OTHER= bar

Table 33.4. Options to Enhance Plots Produced on Graphics Devices

ANNOTATE= *SAS-data-set*	specifies annotate data set with primary vertical axis data units
ANNOTATE2= *SAS-data-set*	specifies annotate data set with secondary vertical axis data units
CTEXT=*color*	specifies color for text
CTEXTSIDE=*color*	specifies color for row labels
CTEXTTOP=*color*	specifies color for column labels
DESCRIPTION='*string*'	specifies description for graphics catalog member
FONT=*font*	specifies font for text
HEIGHT=*value*	specifies text height in percent screen units
HTML=(*variable*)	specifies a variable whose values are URLs to be associated with bars
INFONT=*font*	specifies font for text inside frame
INHEIGHT=*value*	specifies text height in percent screen units for text inside frame
NAME='*string*'	specifies name for graphics catalog member

Table 33.5. Options for Reference Lines

CHREF=*color*	specifies color for HREF= lines
CVREF=*color*	specifies color for VREF= and VREF2= lines
FRONTREF	draws reference lines in front of bars
HREF=*value-list*	requests reference lines perpendicular to horizontal axis
HREFCHAR='*character*'	specifies plot character for HREF= lines
HREFLABELS= ('*label1*'...'*labeln*')	specifies labels for HREF= lines
HREFLABPOS=*n*	specifies position of HREFLABELS= labels
LHREF=*linetype*	specifies line type for HREF= lines
LVREF=*linetype*	specifies line type for VREF= and VREF2= lines
VREF=*value-list*	requests reference lines perpendicular to primary vertical axis
VREF2=*value-list*	requests reference lines perpendicular to secondary vertical axis
VREFCHAR='*character*'	specifies plot character for VREF= lines
VREFLABELS= ('*label1*'...'*labeln*')	specifies labels for VREF= lines
VREF2LABELS= ('*label1*'...'*labeln*')	specifies labels for VREF2= lines
VREFLABPOS=*n*	specifies position of VREFLABELS= and VREF2LABELS= labels

Table 33.6. Options for Comparative Pareto Charts

ANNOKEY	applies annotation only to the key cell
CFRAMESIDE=*color*	specifies frame color for row labels
CFRAMETOP=*color*	specifies frame color for column labels
CLASS=(*variable-list*)	specifies classification variables
CLASSKEY= (*'value1' 'value2'*)	specifies the key cell
CPROP=*color*	specifies color for proportion-of-frequency bar
CTILES=(*variable*)	specifies colors for tile backgrounds
INTERTILE=*value*	specifies distance in percent screen units between tiles
MISSING1	specifies that missing values of the first CLASS= variable be analyzed as a level
MISSING2	specifies that missing values of the second CLASS= variable be analyzed as a level
NCOLS=*n*	specifies number of columns
NOKEYMOVE	suppresses the placement of the key cell in the top left corner
NROWS=*n*	specifies number of rows
ORDER1=*keyword*	specifies the order in which values of the first CLASS= variable are displayed
ORDER2=*keyword*	specifies the order in which values of the second CLASS= variable are displayed
TILELEGEND=(*variable*)	specifies legend for CTILES= colors
TILELEGLABEL=*'label'*	specifies label for TILELEGEND= legend

Table 33.7. Options for Grids

CGRID=*color*	specifies color for GRID lines
CGRID2=*color*	specifies color for GRID2 lines
GRID	adds grid corresponding to primary vertical axis
GRID2	adds grid corresponding to secondary vertical axis
LGRID=*linetype*	specifies line type for GRID lines
LGRID2=*linetype*	specifies line type for GRID2 lines
WGRID=*n*	specifies width of GRID lines
WGRID2=*n*	specifies width of GRID2 lines

Table 33.8. Options for Controlling Axes

ANGLE=*value*	rotates horizontal axis tick mark labels
AXISFACTOR=*value*	specifies distance factor between the tallest bar and the upper frame
CAXIS=*color*	specifies axis color
CAXIS2=*color*	specifies color for secondary vertical axis and tick marks
CFRAME=*color*	specifies color for area enclosed by axes and frame
HOFFSET=*value*	specifies horizontal axis offset in percent screen units
NOCHART	suppresses Pareto chart
NOFRAME	suppresses axis frame
NOHLABEL	suppresses horizontal axis label
NOVLABEL	suppresses primary vertical axis label
NOVLABEL2	suppresses secondary vertical axis label
NOVTICK	suppresses tick marks and tick mark labels for primary vertical axis
NOVTICK2	suppresses tick marks and tick mark labels for secondary vertical axis
SCALE=*keyword*	specifies units in which primary vertical axis is scaled
TURNVLABEL	turns and strings vertically the characters in the primary and secondary vertical axis labels
VAXIS=*value-list*	specifies tick mark values for primary vertical axis
VAXISLABEL='*label*'	labels primary vertical axis
VAXIS2=*value-list*	specifies tick mark values for secondary vertical axis
VAXIS2LABEL='*label*'	labels secondary vertical axis
VOFFSET=*value*	specifies vertical axis offset in percent screen units
WAXIS=*n*	specifies width in pixels for the axes and frame

Table 33.9. Options for Displaying a Sample Size Legend

CFRAMENLEG=*color*	frames the NLEGEND legend and fills the frame with the specified color
NLEGEND	requests sample size legend
NLEGEND='*label*' | (*variable*)	requests sample size legend with specified label

Table 33.10. Options for Displaying Bars

BARLABEL=*keyword*|(*variable-list*)	displays labels for bars
BARLABPOS=*keyword*	specifies position of BARLABEL= option
BARLEGEND=(*variable-list*)	displays legend for CBARS= colors or PBARS= patterns
BARLEGLABEL='*label*'	displays label for BARLEGEND= legend
BARWIDTH=*value*	specifies width (vertical dimension) in percent screen units of the bars
CATLEGLABEL='*label*'	specifies label for Pareto categories legend
CBARLINE=*color*	specifies color for bar outlines
CBARS=*color*	specifies color for bars
CBARS=*variable-list*	specifies variable that provides bar colors
CHIGH(n)=*color*	specifies color for bars with the n highest values
CLOW(n)=*color*	specifies color for bars with the n lowest values
HLLEGLABEL='*label*'	displays label for the legend that describes colors and patterns of highest or lowest bars
INTERBAR=*value*	specifies distance between bars in percent screen units
LABOTHER='*other-label*'	specifies label for "other" category
LAST='*category*'	specifies bottommost category
NOHLLEG	suppresses legend describing colors and patterns of highest or lowest bars
PBARS=*pattern*	specifies pattern for the bars
PBARS=(*variable-list*)	specifies variable that provides bar patterns
PHIGH(n)=*pattern*	specifies pattern for bars with the n highest values
PLOW(n)=*pattern*	specifies pattern for bars with the n lowest values
WBARLINE=*n*	specifies width for bar outlines

Dictionary of Options

The following entries provide detailed descriptions of options you can specify after the slash (/) in the VBAR statement. For example, to request that the bars of your Pareto chart be colored red, use the CBARS= option, as follows:

```
proc pareto data=failure;
   vbar cause / cbars = red ;
run;
```

The marginal notes *Graphics* and *Line Printer* identify options that apply only to charts displayed with graphics devices and line printers, respectively.

ANCHOR=BC | BL | TC | TR
specifies where the Pareto curve is anchored to the leftmost bar on the chart. The following table lists the possible positions.

Keyword	Anchoring position
BC	bottom center
BL	bottom left corner
TC	top center
TR	top right corner

See Output 36.2.1 on page 1062 for an illustration. The default is TR.

ANGLE=*value*
specifies an angle in degrees for rotating the Pareto category labels on the horizontal axis. The *value* is the angle between the baseline of the label and the horizontal axis. See Output 36.1.1 on page 1057 and Output 36.1.2 on page 1057 for an illustration. The *value* must be greater than or equal to −90 and less than 90. If you are using a line printer, only ANGLE=0 and ANGLE=−90 are applicable. If you are using a graphics device and you specify the ANGLE= option, you should also specify a software font with the FONT= option in the VBAR statement or the FTEXT= option in a GOPTIONS statement. The default *value* is zero.

ANNOKEY
specifies that annotation requested with the ANNOTATE= and ANNOTATE2= options is to be applied only to the key cell in a comparative Pareto chart. By default, annotation is applied to all of the cells. *Graphics*

ANNOTATE=*SAS-data-set*
ANNO=*SAS-data-set*
specifies an input data set that contains annotate variables as described in *SAS/GRAPH Software: Reference*. You can use this data set to customize charts. This data set is associated with the primary vertical axis. If the annotation is based on data coordinates, you must use the same units as the primary vertical axis. Features provided in the ANNOTATE= data set are added to every chart produced with the VBAR statement. *Graphics*

ANNOTATE2=*SAS-data-set*
ANNO2=*SAS-data-set*
specifies an input data set that contains annotate variables as described in *SAS/GRAPH Software: Reference*. You can use this data set to customize charts. This data set is associated with the secondary vertical axis. If the annotation is based on data coordinates, you must use the same units as the secondary vertical axis. Features provided in the ANNOTATE2= data set are added to every chart produced with the VBAR statement. *Graphics*

AXISFACTOR=*value*

specifies a factor used in scaling the primary vertical axis. This factor determines (approximately) the ratio of the length of the axis to the length of the tallest bar, and it is used to provide space for the cumulative percent curve. The *value* must be greater than one.

By default, the factor is chosen so that the curve will be anchored at the top right corner of the leftmost bar (see also the ANCHOR= option). If this causes the bars to be flattened excessively, however, a smaller default factor is used.

The AXISFACTOR= option is not applicable if the curve is suppressed with the NOCURVE option.

BARLABEL = VALUE | CMPCT | (*variable***)**

specifies that a label is to be displayed above each bar. If you specify BARLABEL=VALUE, the label indicates the height of the bar in the units used by the primary vertical axis. See Example 36.8 on page 1075 for an illustration.

If you specify BARLABEL=CMPCT, the label indicates the cumulative percent for that bar. An alternative to BARLABEL=CMPCT is the CMPCTLABEL option, which labels points on the cumulative percent curve with the cumulative percents.

If you specify BARLABEL= (*variable*), the label indicates the values of the variable specified in parentheses. The variable can have a formatted length less than or equal to 32. If a format is associated with the variable, then the formatted value is displayed. The values must be consistent within observations corresponding to a particular Pareto category. The variable is saved in the OUT= data set.

BARLABPOS=VFIT | HCENTER | VBAR | HLJUST

specifies the position for labels requested with the BARLABEL= option.

- BARLABPOS=VFIT displays the labels vertically on or above the bars, depending on the available space (the labels are truncated if necessary).
- BARLABPOS=HCENTER centers the labels horizontally above the bars.
- BARLABPOS=VBAR displays the labels vertically on the bars (the labels are truncated if necessary).
- BARLABPOS=HLJUST left-justifies the labels horizontally above the bars.

By default, the labels are horizontally centered above the bars, with a reduction in text height if necessary. Reduction is not applied with the BARLABPOS= options.

BARLEGEND=(*variable-list***)**

Graphics

specifies that a legend is to be added to the chart to explain colors for bars specified with the CBARS=(*variable-list*) option or patterns for bars specified with the PBARS=(*variable-list*) option. The *variable-list* must be enclosed in parentheses even if only one *variable* is specified. See Output 36.4.1 on page 1069 for an illustration.

The values of the BARLEGEND= variable provide the explanatory labels used in the legend. The variable can have a formatted length that does not exceed 32. If a format is associated with the variable, then the formatted value is displayed.

The BARLEGEND= option is not applicable unless you specify CBARS=(*variable-list*) or PBARS=(*variable-list*) or both. In the DATA= data set, the values of the BARLEGEND= variable must be identical in observations for which the value of the CBARS= variable or the PBARS= variable (or the combination of these two values) is the same. This ensures that the legend derived from the BARLEGEND= variable is consistent.

If you specify more than one process variable in the chart statement and a corresponding list of CBARS= or PBARS= variables, you can specify a list of BARLEGEND= variables. The number of BARLEGEND= variables should be less than or equal to the number of process variables. The lists of variables are matched so that the first BARLEGEND= variable is applied to the first process variable and the first CBARS= or PBARS= variable, the second BARLEGEND= variable is applied to the second process variable and the second CBARS= or PBARS= variable, and so forth. If the list of process variables is longer than the *variable-list* in the BARLEGEND= option, the charts for the extra process variables will not display a bar legend.

BARLEGLABEL=*'label'*

specifies the label displayed to the left of the legend created with the BARLEGEND= option. See Output 36.4.1 on page 1069 for an illustration. *Graphics*

The BARLEGLABEL= option is applicable only in conjunction with CBARS= or PBARS= variables. The label can be up to 16 characters and must be enclosed in quotes.

If no label is specified with the BARLEGLABEL= option, the label associated with the BARLEGEND= variable is displayed (unless the label is longer than 16 characters, in which case the variable name is displayed). If the BARLEGLABEL= option is not specified and no label is associated with the BARLEGEND= variable, no legend label is displayed. If both labels are specified, the BARLEGLABEL= label takes precedence over the variable label.

BARWIDTH=*value*

specifies the width of the bars in percent screen units. By default, the bars are as wide as possible. *Graphics*

CATLEGLABEL=*'label'*

specifies a label for the category legend that is added when there is insufficient space to label the categories along the horizontal axis. The *label* can be up to 16 characters and must be enclosed in quotes. The default label is *Categories:*. See Example 36.3 on page 1066 for an illustration. The CATLEGLABEL= option is ignored if it is unnecessary to add the legend.

CAXIS=*color*
CAXES=*color*
CA=*color*

specifies the color for the axis line and tick marks on the chart. The default color is the first color in the device color list. This color is also used for bar outlines and grid lines, unless overridden by the CBARLINE=, CGRID=, or CGRID2= options. *Graphics*

CAXIS2=*color*

Graphics

specifies the color for the tick mark labels and axis label associated with the secondary axis. By default, the color specified with the CTEXT= option (or its default) is used.

CBARLINE=*color*

Graphics

specifies the color for bar outlines. By default, bar outlines are the same color as the axes.

CBARS=*color*

CBARS=(*variable-list***)**

Graphics

specifies how the bars of the Pareto chart are to be colored. You can use one of the following approaches:

- You can specify a single color to be used for all the bars with CBARS=*color*. You can use this option in conjunction with the CHIGH and CLOW options. See Output 36.2.1 on page 1062 for an illustration.
- You can specify a distinct color for each bar (or combination of bars) by providing the colors as values of a CBARS= variable. This variable must be a character variable of length eight. You can use the special value `EMPTY` to indicate that a bar is not to be colored. Note that the variable name must be enclosed in parentheses. You cannot specify a CBARS= variable in conjunction with the CHIGH and CLOW options. See Output 36.3.1 on page 1067 and Output 36.4.1 on page 1069 for examples.

If you specify more than one process variable, you can specify more than one CBARS= variable. The number of CBARS= variables should be less than or equal to the number of process variables. The two lists of variables are paired in order of specification. If a CBARS= variable is not provided for a process variable, the bars for that chart are not colored.

If you specify one or more CBARS= variables, you can also use the BARLEGEND= option to add a legend to the chart that explains the significance of each color. Furthermore, you can use the PBARS= option to specify patterns in conjunction with the CBARS= option. See Output 36.4.1 on page 1069 and Output 36.5.1 on page 1070 for examples.

CCONNECT=*color*

Graphics

specifies the color for the line segments connecting the points on the cumulative percent curve. You can specify the color for the points themselves with the COLOR= option in the SYMBOL statement; this is the default color.

CFRAME=*color*

Graphics

specifies the color for filling the area enclosed by the axes and the frame. By default, this area is not filled. The CFRAME= option cannot be used in conjunction with the NOFRAME option or the CTILES= option.

CFRAMENLEG=*color*

specifies that the legend requested with the NLEGEND option be framed and that the frame be filled with the color indicated. If you specify CFRAMENLEG=EMPTY, a frame is drawn but is not filled with a color. See Figure 33.4 on page 967 and Output 36.1.4 on page 1059 for illustrations. *Graphics*

CFRAMESIDE=*color*

specifies the color for filling the frame area for the row labels displayed along the left side of a comparative Pareto chart requested with the CLASS= option. If a label is associated with the classification variable, this color is also used to fill the frame area for this label. By default, these areas are not filled. *Graphics*

CFRAMETOP=*color*

specifies the color for filling the frame area for the column labels displayed across the top of a comparative Pareto chart requested with the CLASS= option. If a label is associated with the classification variable, this color is also used to fill the frame area for this label. By default, these areas are not filled. *Graphics*

CGRID=*color*

specifies the color for grid lines requested with the GRID option. The default color is the first color in the device color list. By default, grid lines are the same color as the axes. If you specify the CGRID= option, you do not need to specify the GRID option. *Graphics*

CGRID2=*color*

specifies the color for grid lines requested with the GRID2 option. The default color is the first color in the device color list. By default, grid lines are the same color as the axes. If you specify the CGRID2= option, you do not need to specify the GRID2 option. *Graphics*

CHIGH(n**)=***color*

specifies the color used to fill the bars with the n highest values. You cannot use the CHIGH option in conjunction with a CBARS= variable, but you can use the CHIGH(n)= option together with the CLOW(n)= and CBARS=*color* options. See Output 36.3.1 on page 1067 for an illustration. By default, the bars are empty. *Graphics*

CHREF=*color*

specifies the color for lines requested with the HREF= option. The default is the first color in the device color list. *Graphics*

CLASS=*variable*

CLASS=(*variable1 variable2***)**

creates a comparative Pareto chart using the levels of the *variables*. You must enclose two *variables* in parentheses. See Example 36.1 on page 1056 and Example 36.2 on page 1060.

If you specify a single CLASS= *variable*, the observations in the input data set are classified by the formatted values (levels) of the *variable*. A Pareto chart is created for the process variable values in each level, and these component charts (referred to as cells) are arranged in an array. The cells are labeled with the levels, and uniform horizontal and vertical axes are used to facilitate comparisons.

If you specify two CLASS= *variables*, the observations in the input data set are cross-classified by the values (levels) of the *variables*. A Pareto chart is created for the process variable values in each cell of the cross-classification, and these charts are arranged in a *matrix*. The levels of the first CLASS= *variable* label the rows, and the levels of the second CLASS= *variable* label the columns. Uniform horizontal and vertical axes are used to facilitate comparisons.

Note that the array or matrix comparative Pareto chart is displayed only if the chart is produced on a graphics device. If the chart is produced on a line printer, the cells are printed separately.

The CLASS= *variables* can be numeric or character. The maximum length of a character *variable* is 32. If a format is associated with a CLASS= *variable*, the formatted values determine the levels. Only the first 32 characters of the formatted value are used to determine the levels. You can specify whether missing values are to be treated as a level with the MISSING1 and MISSING2 options.

If a label is associated with a CLASS= *variable*, the label is displayed on the chart. On charts produced with a graphics device, the *variable* label is displayed parallel to the column (or row) labels. On charts produced with a line printer, the *variable* label is displayed at the top of the chart.

CLASSKEY=*'value'*
CLASSKEY=(*'value1' 'value2'***)**

specifies the *key cell* in a comparative histogram requested with the CLASS= option. The bin size and midpoints are first determined for the key cell, and then the midpoint list is extended to accommodate the data ranges for the remaining cells. Thus the choice of the key cell determines the uniform horizontal axis used for all cells.

If you specify CLASS=*variable*, you can specify CLASSKEY=*'value'* to identify the key cell as the level for which *variable* is equal to *value*. The *value* can have up to 32 characters, and you must specify a formatted *value*. By default, the levels are sorted in the order determined by the ORDER1= option, and the key cell is the level that occurs first in this order. The cells are displayed in this order from top to bottom (or left to right), and consequently the key cell is displayed at the top or at the left. If you specify a different key cell with the CLASSKEY= option, this cell is displayed at the top or at the left unless you also specify the NOKEYMOVE option.

If you specify CLASS=(*variable1 variable2*), you can specify CLASSKEY=(*'value1' 'value2'*) to identify the key cell as the level for which *variable1* is equal to *value1* and *variable2* is equal to *value2*. Here, *value1* and *value2* must be formatted values, and they must be enclosed in quotes. By default, the levels of *variable1* are sorted in the order determined by the ORDER1= option, and then within each of these levels, the levels of *variable2* are sorted in the order determined by the ORDER2= option. The default key cell is the combination of levels of *variable1* and *variable2* that occurs first in this order. The cells are displayed in order of *variable1* from top to bottom and in order of *variable2* from left to right. Consequently, the default key cell is displayed in the upper left corner. If you specify a different key cell with the CLASSKEY= option, this cell is displayed in the upper left corner unless you also specify the NOKEYMOVE option.

For an example of the use of the CLASSKEY= option, see Output 36.1.3 on page 1058.

CLOW(n**)=***color*

specifies the color used to fill the bars with the n lowest values. You cannot use the CLOW(n)= option in conjunction with a CBARS=*variable*, but you can use the CLOW(n)= option together with the CBARS=*color* and CHIGH options. See Output 36.3.1 on page 1067 for an illustration of the CHIGH(n)= option. By default, the bars are empty. *Graphics*

CMPCTLABEL

labels points on the cumulative percent curve with their values. By default, the points are not labeled.

CONNECTCHAR=*'character'*
CCHAR=*'character'*

specifies the plot character for line segments that connect points on the cumulative percent curve. The default character is a plus sign (+). *Line Printer*

COTHER=*color*

specifies the color for the bar defined by the OTHER= option. By default, this bar is not filled with a color. The COTHER= option is not applicable unless a CBARS=*variable* is specified. *Graphics*

CPROP=*color*

specifies the color for a proportion-of-frequency bar that is displayed horizontally across the top of each tile in a comparative Pareto chart. The length of the bar relative to the width of the tile indicates the proportion of the total frequency count in the chart that is represented by the tile. You can use the bars to visualize the distribution of frequency count by tile. See Output 36.1.4 on page 1059 for an illustration. *Graphics*

The CPROP= option provides a graphical alternative to the NLEGEND options, which display the actual count. The CPROP= option is applicable only with comparative Pareto charts. Empty bars are displayed if you specify CPROP=EMPTY. Bars are not displayed if the CPROP= option is not specified.

CTEXT=*color*
CT=*color*

specifies the color for text, such as tick mark labels, axis labels, and legends. The default is the value specified for the CTEXT= option in the GOPTIONS statement. *Graphics*

CTEXTSIDE=*color*

specifies the color for row labels displayed along the left side of a comparative Pareto chart requested with the CLASS= option. The default color is the color specified with the CTEXT= option in the VBAR statement or the CTEXT= option in the GOPTIONS statement. *Graphics*

CTEXTTOP=*color*

specifies the color for column labels displayed across the top of a comparative Pareto chart requested with the CLASS= option. The default color is the color specified with the CTEXT= option in the VBAR statement or the CTEXT= option in the GOPTIONS statement. *Graphics*

CTILES=(*variable***)**

Graphics

specifies a character variable of length eight whose values are the fill colors for the tiles in a comparative Pareto chart. The CTILES= option generalizes the CFRAME= option, which provides a single color for all of the tiles. The *variable* must be enclosed in parentheses. The values of the *variable* must be identical for all observations with the same level of the CLASS= variables. You can use the same color to fill more than one tile. You can use the special value `EMPTY` to indicate that a tile is not to be filled.

The CTILES= option cannot be used in conjunction with the NOFRAME option or the CFRAME= option. You can use the TILELEGEND= option in conjunction with the CTILES= option to add an explanatory legend for the CTILES= colors at the bottom of the chart. See Output 36.5.1 on page 1070 for an illustration. By default, the tiles are not filled.

CVREF=*color*

Graphics

specifies the color for lines requested with the VREF= and VREF2= options. The default color is the first color in the device color list.

DESCRIPTION=*'string'*
DES=*'string'*

Graphics

specifies a descriptive string, up to 40 characters, that appears in the description field of the PROC GREPLAY master menu.

FONT=*font*

Graphics

specifies a software font for text used in labels and legends. The FONT= option takes precedence over the FTEXT= option in the GOPTIONS statement.

FREQ=*variable*

specifies a frequency variable whose value provides the counts (numbers of occurrences) of the values of the process variable. Specifying a FREQ= variable is equivalent to replicating the observations in the input data set. The FREQ= variable must be a numeric variable with nonnegative integer values. See "Creating a Pareto Chart Using Frequency Data" on page 966 for an illustration. If you specify more than one process variable in the chart statement, the FREQ= variable values are used with each process variable. If you do not specify a FREQ= variable, each value of the process variable is counted exactly once.

FRONTREF

Graphics

draws reference lines requested with the HREF= and VREF= options in front of the bars on the Pareto chart. By default, reference lines are drawn behind the bars and can be obscured by them.

GRID

adds a grid to the Pareto chart corresponding to the primary vertical axis. Grid lines are horizontal lines positioned at tick marks on the primary vertical axis. The lines are useful for comparing the heights of the bars.

GRID2
adds a grid to the Pareto chart corresponding to the secondary vertical axis. Grid lines are horizontal lines positioned at tick marks on the secondary vertical axis. The lines are useful for reading the cumulative percent curve.

HEIGHT=*value*
specifies the height in percent screen units of text for labels and legends. This option should be used only in conjunction with the FONT= option. The HEIGHT= option takes precedence over the HTEXT= option in a GOPTIONS statement. *Graphics*

HLLEGLABEL=*'label'*
specifies a label displayed to the left of the legend that is automatically created when you use a combination of the CHIGH, CLOW, PHIGH, and PLOW options. See Output 36.3.1 on page 1067 for an illustration. The *label* can be up to 16 characters and must be enclosed in quotes. The default *label* is *Bars:*. *Graphics*

HOFFSET=*value*
specifies the length in percent screen units of the offset at both ends of the horizontal axis. You can eliminate the offset by specifying HOFFSET=0. *Graphics*

HREF=*'value-list'*
specifies where reference lines perpendicular to the horizontal (Pareto category) axis are to appear on the chart. Character values can be up to 32 characters and must be enclosed in quotes. The values must be values of the process variable even when the bars are numbered and a category legend is introduced.

HREFCHAR=*'character'*
specifies the plot character used to form the lines requested with the HREF= option. The default character is a vertical bar (|). *Line Printer*

HREFLABELS=*'label1'...'labeln'*
specifies labels for the lines requested with the HREF= option. The number of labels must equal the number of lines requested. Labels can be up to 16 characters and must be enclosed in quotes.

HREFLABPOS=*n*
specifies the vertical positioning of the HREFLABELS= labels. HREFLABPOS=1 positions the labels along the top of the chart. HREFLABPOS=2 staggers the labels from top to bottom. HREFLABPOS=3 positions the labels along the bottom. By default, HREFLABPOS=1.

HTML=*variable*
specifies URLs as values of the specified character variable (or formatted values of a numeric variable). These URLs are associated with bars on the Pareto chart when high resolution graphics output is directed into HTML. The value of the HTML= variable should be the same for each observation with a given value of the subgroup variable.

INFONT=*font*

Graphics

specifies a software font for text used inside the frame of the chart, such as sample size legends. The INFONT= option takes precedence over the FONT= option and the FTEXT= option in the GOPTIONS statement.

INHEIGHT=*value*

Graphics

specifies the height in percent screen units of text used inside the frame of the chart, such as sample size legends and bar labels. This option should be used in conjunction with the INFONT= option.

INTERBAR=*value*

specifies the distance in percent screen units between bars on the chart. By default, the bars are contiguous. See Figure 33.4 on page 967 for an illustration.

INTERTILE=*value*

Graphics

specifies the distance in horizontal percent screen units between tiles (cells) in a comparative Pareto chart. By default, the tiles are contiguous. See Output 36.1.3 on page 1058 for an illustration.

LABOTHER = *'other-label'*

is used in conjunction with the BARLABEL=(variable) option and specifies a label for the "other" category that is optionally specified with the OTHER= option.

LAST=*'category'*

specifies that the bar corresponding to the *category* is to be displayed at the right end of the chart regardless of the percent associated with this category. The *category* must be a formatted value of the process variable and must be enclosed in quotes. The *category* can be up to 32 characters. See Figure 33.6 on page 969 for an illustration.

LGRID=*line-type*

Graphics

specifies the line type for the grid requested with the GRID option. The default *line-type* is 1, which produces a solid line. If you specify the LGRID= option, you do not need to specify the GRID option.

LGRID2=*line-type*

Graphics

specifies the line type for the grid requested with the GRID2 option. The default *line-type* is 1, which produces a solid line. If you specify the LGRID2= option, you do not need to specify the GRID2 option.

LHREF=*line-type*

LH=*line-type*

Graphics

specifies the line type for lines requested with the HREF= option. The default *line-type* is 2, which produces a dashed line.

LOTHER=*'label'*

specifies a label for the bar defined with the OTHER= option. This label appears in the legend created with the BARLEGEND= option. The *label* must be enclosed in quotes and can be up to 32 characters. The default *label* is the value specified with the OTHER= option. The LOTHER= option is applicable only when a BARLEGEND= variable is specified.

LVREF=*line-type*
LV=*line-type*

specifies the line type for lines requested with the VREF= and VREF2= options. See Output 36.2.3 on page 1064 for an illustration. The default *line-type* is 2, which produces a dashed line. *Graphics*

MAXCMPCT=*percent*

specifies that only the Pareto categories with the n highest frequency counts are to be displayed, where the sum of the n corresponding percents is less than or equal to the specified *percent*. For example, if you specify

```
proc pareto data=failure;
   vbar cause / maxcmpct = 90 ;
```

the chart displays only the n most frequently occurring categories that account for no more than 90 percent of the total frequency.

You can use the OTHER= option in conjunction with the MAXCMPCT= option to create and display a new category that combines those categories that are not selected with the MAXCMPCT= option. For example, if you specify

```
proc pareto data=failure;
   vbar cause / maxcmpct = 90
                other    = 'Others' ;
```

the chart displays the categories that account for no more than 90 percent of the total frequency, together with a category labeled *Others* that merges the remaining categories. The MAXCMPCT= option is an alternative to the MINPCT= and MAXNCAT= options.

MAXNCAT=*n*

specifies that only the Pareto categories with the n highest frequencies are to be displayed. For example, if you specify

```
proc pareto data=failure;
   vbar cause / maxncat = 20 ;
```

the chart displays only the categories with the 20 highest frequencies. If the total number of categories is less than 20, all the categories are displayed.

You can use the OTHER= option in conjunction with the MAXNCAT= option to create and display a new category that combines those categories that are not selected with the MAXNCAT= option. For example, if you specify

```
proc pareto data=failure;
   vbar cause / maxncat = 20
                other= 'Others' ;
```

the chart displays the categories with the 19 highest frequencies, together with a category labeled *Others* that merges the remaining categories. See Figure 33.6 on page 969 for another illustration.

The MAXNCAT= option is an alternative to the MINPCT= and MAXCMPCT= options.

MINPCT=*percent*
specifies that only the Pareto categories with frequency percents greater than or equal to the specified *percent* are to be displayed. For example, if you specify

```
proc pareto data=failure;
   vbar cause / minpct = 5 ;
```

the chart displays only those categories with at least five percent of the total frequency.

You can use the OTHER= option in conjunction with the MINPCT= option to create and display a new category that combines those categories that are not selected with the MINPCT= option. The merged category created by the OTHER= option is displayed even if its total percent is less than the *percent* specified with the MINPCT= option. For example, if you specify

```
proc pareto data=failure;
   vbar cause / minpct = 5
                other  = 'Others' ;
```

the chart displays the categories with percents greater than or equal to five percent, together with a category labeled *Others* that merges the remaining categories.

The MINPCT= option is an alternative to the MAXNCAT= and MAXCMPCT= options.

MISSING
specifies that missing values of the process variable are to be treated as a Pareto category represented with a bar on the chart. If the process variable is a character variable, a missing value is defined as a blank internal (unformatted) value. If the process variable is numeric, a missing value is defined as any of the SAS missing values. If you do not specify the MISSING option, missing values are excluded from the analysis.

MISSING1
specifies that missing values of the first CLASS= variable are to be treated as a level of the CLASS= variable. If the first CLASS= variable is a character variable, a missing value is defined as a blank internal (unformatted) value. If the first CLASS= variable is numeric, a missing value is defined as any of the SAS missing values. If you do not specify MISSING1, observations in the DATA= data set for which the first CLASS= variable is missing are excluded from the analysis.

MISSING2
specifies that missing values of the second CLASS= variable are to be treated as a level of the CLASS= variable. If the second CLASS= variable is a character variable, a missing value is defined as a blank internal (unformatted) value. If the second CLASS= variable is numeric, a missing value is defined as any of the SAS missing values. If you do not specify MISSING2, observations in the DATA= data set for which the second CLASS= variable is missing are excluded from the analysis.

NAME=*'string'*
specifies a name for the chart, up to eight characters, that appears in the PROC GREPLAY master menu. The default name is 'PARETO'. *Graphics*

NCOLS=*n*
NCOL=*n*
specifies the number of columns in a comparative Pareto chart. You can use the NCOLS= option in conjunction with the NROWS= option. See Output 36.2.3 (page 1064) and Output 36.2.4 (page 1065) for an illustration. By default, NCOLS=1 and NROWS=2 if one CLASS= variable is specified, and NCOLS=2 and NROWS=2 if two CLASS= variables are specified.

NLEGEND
NLEGEND='*label*'
NLEGEND=(*variable***)**
requests a sample size legend and specifies its form as follows:

- If you specify the NLEGEND option, the form is *N*=*n*, where n is the total count for the Pareto categories. In a comparative Pareto chart, a legend is displayed in each tile, and n is the total count for that particular cell. See Output 36.2.1 on page 1062 for an illustration.
- If you specify the NLEGEND='*label*' option, the form is *label*=*n*, where n is the total count for the Pareto categories. The label can be up to 32 characters and must be enclosed in quotes. For an illustration, see Figure 33.4 on page 967 or Output 36.1.4 on page 1059.
- If you specify the NLEGEND=(*variable*) option, the legend is the value of the *variable*, which must be a variable in the DATA= data set whose formatted length does not exceed 32. If a format is associated with the variable, then the formatted value is displayed. This option is intended for use with comparative Pareto charts and enables you to display a customized legend inside each tile (this legend need not provide total count). It is assumed that the values of the *variable* are identical for all observations in a particular class.

By default, the legend is placed in the upper-left corner of the chart. If the NOCURVE option is specified, the legend is placed in the upper-right corner of the chart. You can use the CFRAMENLEG= option to frame the sample size legend. No legend is displayed if you do not specify an NLEGEND option.

NOCHART
suppresses the creation of a Pareto chart. This option is useful when you are simply creating an output data set.

NOCURVE
suppresses the display of the cumulative percent curve and the secondary vertical axis. Compare Output 36.2.1 (page 1062) and Output 36.2.2 (page 1063) for an illustration.

NOFRAME
suppresses the frame that is drawn around the chart by default. The NOFRAME option cannot be specified in conjunction with the CFRAME= or CTILES= options.

NOHLABEL
suppresses the label for the horizontal axis. This is useful for avoiding clutter in situations where the meaning of the horizontal axis is apparent from the labels for the Pareto categories. See Output 36.2.2 on page 1063 for an illustration.

NOHLLEG
suppresses the legend generated by the CHIGH(n)=, CLOW(n)=, PHIGH(n)=, and PLOW(n)= options.

NOKEYMOVE
Graphics
suppresses the rearrangement of cells within a comparative Pareto chart that occurs when you use the CLASSKEY= option. The key cell appears in the top left corner of a comparative Pareto chart unless you use the CLASSKEY= option together with the NOKEYMOVE option.

NOVLABEL
suppresses the label for the primary vertical axis.

NOVLABEL2
suppresses the label for the secondary vertical axis. This is useful for avoiding clutter on comparative Pareto charts.

NOVTICK
suppresses the primary vertical axis label, tick marks, and tick mark labels.

NOVTICK2
suppresses the secondary vertical axis label, tick marks, and tick mark labels.

NROWS=n
NROW=n
Graphics
specifies the number of rows in a comparative Pareto chart. You can use the NROWS= option in conjunction with the NCOLS= option. See Output 36.2.3 on page 1064 and Output 36.2.4 on page 1065 for an illustration. By default, NROWS=2.

ORDER1=INTERNAL | FORMATTED | DATA | FREQ
specifies the display order for the values of the first CLASS= variable. The levels of the first CLASS= variable are always constructed using the formatted values of the variable, and the formatted values are always used to label the rows (columns) of a comparative Pareto chart.

If you specify ORDER1=INTERNAL, the rows (columns) are displayed from top to bottom (left to right) in increasing order of the internal, or unformatted, values of the first CLASS= variable. If there are two or more distinct internal values with the same formatted value, the order is determined by the internal value that occurs first in the input data set. For example, suppose that you use a numeric CLASS= variable called DAY (with values 1, 2, and 3) to create a one-way comparative Pareto chart. Suppose also that you use the FORMAT procedure to associate the formatted values 1 = 'Wednesday', 2 = 'Thursday', and 3 = 'Friday' with the variable DAY. If you specify ORDER1=INTERNAL, the rows of the comparative chart will appear in chronological order (*Wednesday*, *Thursday*, *Friday*) from top to bottom.

If you specify ORDER1=FORMATTED, the rows (columns) are displayed from top to bottom (left to right) in increasing order of the formatted values of the first CLASS= variable. For instance, in the previous illustration, if you specify ORDER1=FORMATTED, the rows will appear in alphabetical order (*Friday*, *Thursday*, *Wednesday*) from top to bottom.

If you specify ORDER1=DATA, the rows (columns) are displayed from top to bottom (left to right) in the order in which the values of the first CLASS= variable first appear in the input data set.

If you specify ORDER1=FREQ, the rows (columns) are displayed from top to bottom (left to right) in order of *decreasing* frequency count. If two or more classes have the same frequency count, the order is determined by the formatted values.

By default, ORDER1=INTERNAL.

ORDER2=INTERNAL | FORMATTED | DATA | FREQ
specifies the display order for the values of the second CLASS= variable. The levels of the second CLASS= variable are always constructed using the formatted values of the variable, and the formatted values are always used to label the columns of a two-way comparative Pareto chart.

The PARETO procedure determines the layout of a two-way comparative Pareto chart by first using the ORDER1= option to obtain the order of the rows from top to bottom (recall that ORDER1=INTERNAL by default). Then the ORDER2= option is applied to the observations corresponding to the first row to obtain the order of the columns from left to right. If any columns remain unordered (that is, the categories are unbalanced), the ORDER2= option is applied to the observations in the second row, and so on until all the columns have been ordered.

The values of the ORDER2= option are interpreted as described for the ORDER1= option. By default, ORDER2=INTERNAL.

OTHER=*'category'*
specifies a new category that merges all categories not selected with the MAXNCAT=, MINPCT=, or MAXCMPCT= options. See Figure 33.6 on page 969 for an illustration.

The *category* should be specified as a formatted value of the process variable. The *category* can be up to 32 characters and must be enclosed in quotes. If you specify

an OUT= data set, you should also specify an internal value corresponding to the *category* with the OTHERCVAL= option or the OTHERNVAL= option.

The OTHER= option is not applicable unless you specify the MAXNCAT=, MINPCT=, or MAXCMPCT= option. You can use the COTHER=, LOTHER=, POTHER=, OTHERCVAL=, and OTHERNVAL= options with the OTHER= option.

OTHERCVAL=*'value'*
specifies the internal (unformatted) value for a character process variable in the OUT= data set that corresponds to the category created with the OTHER= option. The *category* can be up to 32 characters and must be enclosed in quotes.

The OTHERCVAL= option is not applicable unless you specify the OTHER= and OUT= options. If you specify the OTHER= option but not the OTHERCVAL= option, the default *value* is the *value* specified with the OTHER= option.

OTHERNVAL=*value*
specifies the internal (unformatted) value for a numeric process variable in the OUT= data set that corresponds to the category created with the OTHER= option. The OTHERNVAL= option is not applicable unless you specify the OTHER= and OUT= options. If you specify the OTHER= option but not the OTHERNVAL= option, *value* is assigned a missing value.

OUT=*SAS-data-set*
creates an output data set that contains the information displayed in the Pareto chart. This is useful if you want to create a report to accompany your chart. See Example 36.8 on page 1075 for an illustration.

PBARS=*pattern*
PBARS=(*variable-list***)**
Graphics
specifies pattern fills for the bars. You can use one of two approaches:

- You can specify a single pattern to be used for all the bars with the PBARS=*pattern* option. You can use this option in conjunction with the PHIGH and PLOW options. See Output 36.2.1 on page 1062 for an illustration.
- You can specify a distinct pattern for *each* bar (or combination of bars) by providing the patterns as values of a PBARS= variable. For example, you might use the solid pattern (`S`) to indicate severe problems and the empty pattern (`E`) for all other problems. The variable must be a character variable of length eight, and the variable name must be enclosed in parentheses. You cannot specify a PBARS= variable in conjunction with the PHIGH and PLOW options. See Output 36.4.1 on page 1069 and Output 36.5.1 on page 1070 for illustrations.

If you specify more than one process variable in the chart statement, you can provide more than one PBARS= variable. The number of PBARS= variables should be less than or equal to the number of process variables. The two lists of variables are paired in order of specification. If a PBARS= variable is not provided for a process variable, the bars for that chart are not filled.

If you specify one or more variables with the PBARS= option, you can also use the BARLEGEND= option to add a legend to the chart that explains the significance

of each pattern. Furthermore, you can use the CBARS= option to specify colors in conjunction with the PBARS= option. See Output 36.4.1 on page 1069 and Output 36.5.1 on page 1070 for illustrations.

PHIGH(*n*)=*pattern*

specifies the pattern used to fill the bars with the n highest values. You cannot specify the PHIGH option in conjunction with a PBARS= variable, but you can specify the PHIGH(n)= option together with the PLOW(n)= and PBARS=*pattern* options. See Output 36.3.1 on page 1067 for an illustration. By default, the bars are empty. *Graphics*

PLOW(*n*)=*pattern*

specifies the pattern used to fill the bars with the n lowest values. You cannot specify the PLOW option in conjunction with a PBARS= variable, but you can use the PLOW(n)= option together with the PHIGH(n)= and PBARS=*pattern* options. See Output 36.3.1 on page 1067 for an illustration of the PHIGH(n)= option. By default, the bars are empty. *Graphics*

POTHER=*pattern*

specifies the pattern used for the bar defined by the OTHER= option. By default, this bar is empty. The POTHER= option is not applicable unless a PBARS= variable is specified. *Graphics*

SCALE=PERCENT | COUNT | WEIGHT

specifies the scale for the primary vertical axis.

If you specify SCALE=PERCENT, the scale is percent of total frequency. If a WEIGHT= variable is used, the scale is percent of total weight.

If you specify SCALE=COUNT, the scale is counts. See Output 36.1.4 on page 1059 for an illustration. This option is not applicable if a WEIGHT= variable is used. You can specify SCALE=FREQUENCY instead of SCALE=COUNT.

If you specify SCALE=WEIGHT, the vertical axis is scaled in the same units as the WEIGHT= variable. This option is not applicable unless you use a WEIGHT= variable.

By default, SCALE=PERCENT. See Output 36.5.1 on page 1070 for an example. Regardless of how SCALE= is specified, the secondary axis is scaled in cumulative percent units.

SYMBOLCHAR=*'character'*

specifies the plot character for points on the cumulative percent curve. The default character is an asterisk (*). *Line Printer*

TILELEGEND=(*variable*)

specifies a variable used to add a legend for CTILES= colors. The variable can have a formatted length less than or equal to 32. If a format is associated with the variable, then the formatted value is displayed. The TILELEGEND= option must be used in conjunction with the CTILES= option for filling the tiles in a comparative Pareto chart. If CTILES= is specified and TILELEGEND= is not specified, a color legend is not displayed. *Graphics*

The values of the CTILES= and TILELEGEND= variables should be consistent for all observations with the same level of the CLASS= variables. The value of the TILELEGEND= variable is used to identify the corresponding color value of the CTILES= variable in the legend. See Output 36.5.1 on page 1070 for an illustration.

TILELEGLABEL=*'label'*

Graphics

specifies a label displayed to the left of the legend that is created when you specify a TILELEGEND= variable. The *label* can be up to 16 characters and must be enclosed in quotes. The default *label* is *Tiles:*. See Output 36.5.1 on page 1070 for an illustration.

TURNVLABEL
TURNVLABELS

Graphics

turns and strings out vertically the characters in the labels for the primary and secondary vertical axes. This happens by default when a hardware font is used.

VAXIS=*value-list*

specifies tick mark values for the primary vertical axis. The values must be equally spaced and in increasing order, and the first *value* must be zero. You must scale the values in the same units as the bars (see the SCALE= option), and the last *value* must be greater than or equal to the height of the largest bar.

VAXISLABEL=*'label'*

specifies a label, up to 40 characters, for the primary vertical axis. The default label depends on the value of the SCALE= option, or it is the label associated with the WEIGHT= variable.

VAXIS2=*value-list*

specifies tick mark values for the secondary vertical axis. The values must be equally spaced and in increasing order, and the first *value* must be zero. You must scale the values in percent units, and the last *value* must be greater than or equal to 100.

VAXIS2LABEL=*'label'*

specifies a label, up to 40 characters, for the secondary vertical axis. The default label is *Cumulative Percent* or *Cm Pct*, depending on the space available.

VOFFSET=*value*

Graphics

specifies the length in percent screen units of the offset at the upper end of the primary vertical axis.

VREF=*value-list*

specifies where reference lines perpendicular to the primary vertical axis are to appear on the chart. You must specify the values in the same units used to scale the primary axis. By default, the primary axis is scaled in percent units, but you can specify other units with the SCALE= option. See Output 36.2.3 on page 1064 for an illustration.

VREF2=*value-list*

specifies where reference lines perpendicular to the secondary vertical axis are to appear on the chart. You must specify the values in cumulative percent units.

VREFCHAR=*'character'*

specifies the character used to form the lines requested with the VREF= and VREF2= options. The default character is a dash (-). *Line Printer*

VREFLABELS=*'label1'...'labeln'*

specifies labels for the lines requested with the VREF= option. The number of labels must equal the number of lines requested. Enclose the labels in quotes. Labels can be up to 16 characters.

VREF2LABELS=*'label1'...'labeln'*

specifies labels for the lines requested with the VREF2= option. The number of labels must equal the number of lines requested. Enclose the labels in quotes. Labels can be up to 16 characters.

VREFLABPOS=*n*

specifies the vertical positioning of the VREFLABELS= and VREF2LABELS= labels. If you specify VREFLABPOS=1, the labels are positioned at the left of the chart, and if you specify VREFLABPOS=2, the labels are positioned at the right. By default, n=1.

WAXIS=*n*

specifies the line thickness (in pixels) for the axes and frame. By default, $n = 1$. This thickness is also used for bar outlines and grid lines, unless overridden by the WBARLINE=, WGRID=, or WGRID2= options. *Graphics*

WBARLINE=*n*

specifies the width for bar outlines. By default, bar outlines are the same width as the axes.

WEIGHT=*variable-list*

specifies weight variables used to construct weighted Pareto charts. The WEIGHT= variables are paired with the process variables in order of specification. The WEIGHT= variables must be numeric, and their values must be nonnegative (noninteger values are permitted). If a WEIGHT= variable is not provided for a process variable, the weights applied to that process variable are assumed to be one. See "Weighted Pareto Charts" on page 1050 for computational details.

A WEIGHT= variable is particularly useful for carrying out a Pareto analysis based on *cost* rather than frequency of occurrence. See Example 36.8 on page 1075 for an illustration.

WGRID=*n*

specifies the width of the primary chart grid lines. By default, grid lines are the same width as the axes. If the WGRID= option is specified the GRID option is not required.

WGRID2=*n*

specifies the width of the secondary chart grid lines. By default, grid lines are the same width as the axes. If the WGRID2= option is specified the GRID2 option is not required.

Chapter 34
HBAR Statement

Chapter Contents

OVERVIEW 999

GETTING STARTED 999
Creating a Pareto Chart from Raw Data 999
Creating a Pareto Chart Using Frequency Data 1001
Restricting the Number of Pareto Categories 1003

SYNTAX 1005
Summary of Options 1005
Dictionary of Options 1010

Chapter 34
HBAR Statement

Overview

The HBAR statement creates a Pareto chart with horizontal bars representing the frequencies of problems in a process or operation. The HBAR statement is available only with high resolution graphics, so it cannot be used when the LINEPRINTER option is specified on the PROC PARETO statement.

A horizontal Pareto chart has one vertical axis on which the Pareto categories are listed. The primary horizontal axis appears at the top of the chart and is used to read the lengths of the bars on the chart. The secondary horizontal axis is at the bottom of the chart and is used to read the cumulative percent curve.

Getting Started

The examples in this section illustrate basic features of the HBAR statement. Complete syntax for the HBAR statement is presented in the "Syntax" section on page 1005.

Creating a Pareto Chart from Raw Data

In the fabrication of integrated circuits, common causes of failures include improper doping, corrosion, surface contamination, silicon defects, metallization, and oxide defects. The causes of 31 failures were recorded in a SAS data set called FAILURE1.

```
data failure1;
   length cause $ 16 ;
   label  cause = 'Cause of Failure' ;
   input  cause $ 3-18 ;
   datalines;
  Corrosion
  Oxide Defect
  Contamination
  Oxide Defect
  Oxide Defect
  Miscellaneous
  Oxide Defect
  Contamination
  Metallization
  Oxide Defect
  Contamination
  Contamination
  Oxide Defect
  Contamination
```

```
  Contamination
  Contamination
  Corrosion
  Silicon Defect
  Miscellaneous
  Contamination
  Contamination
  Contamination
  Miscellaneous
  Contamination
  Contamination
  Doping
  Oxide Defect
  Oxide Defect
  Metallization
  Contamination
  Contamination
;
run;
```

Each of the 31 observations corresponds to a different circuit, and the value of CAUSE provides the cause for the failure. These are raw data in the sense that there is more than one observation with the same value of CAUSE, and the observations are not sorted by CAUSE. The following statements produce a basic Pareto chart for the failures:

```
proc pareto data=failure1;
   hbar cause;
run;
```

The PARETO procedure is invoked with the first statement, referred to as the PROC statement. You specify the process variable to be analyzed in the HBAR statement.

The Pareto chart is shown in Figure 34.1.

The procedure has classified the values of CAUSE into seven distinct categories (levels). The bars represent the percent of failures in each category, and they are arranged in decreasing order. Thus, the most frequently occurring category is *Contamination*, which accounts for 45% of the failures. The Pareto curve indicates the cumulative percent of failures from top to bottom; for example, *Contamination* and *Oxide* together account for 71% of the failures.

If there is sufficient space, the procedure labels the bars along the vertical axis as in Figure 34.1. Otherwise the procedure numbers the bars from top to bottom and adds a legend identifying the categories.

A category legend is likely to be introduced when

- the number of categories is large
- the category labels are lengthy. Category labels can be up to 32 characters.

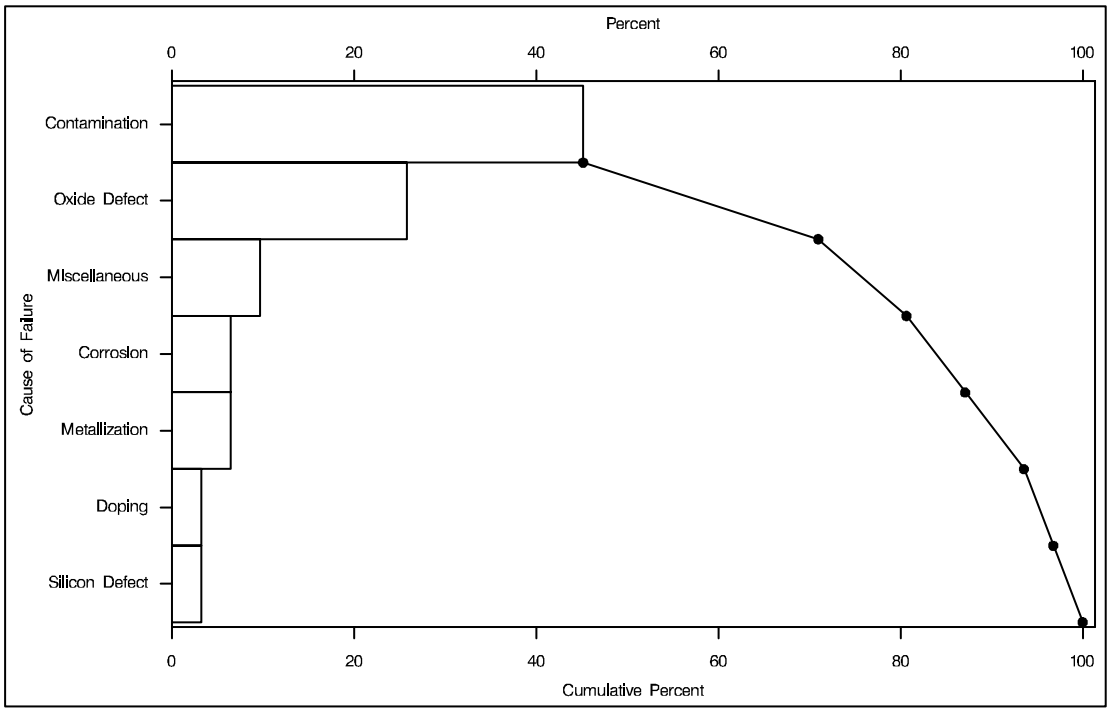


Figure 34.1. Pareto Chart for IC Failures in the Data Set FAILURE1

- a large text height is used. You can specify the height with the HEIGHT= option in the HBAR statement or with the HTEXT= option in a GOPTIONS statement.

Creating a Pareto Chart Using Frequency Data

In some situations, a count (frequency) is available for each category, or you can compress a large data set by creating a frequency variable for the categories before applying the PARETO procedure.

For example, you can use the FREQ procedure to obtain the compressed data set FAILURE2 from the data set FAILURE1.

```
proc freq data=failure1;
   tables cause / noprint out=failure2;
run;

proc print data=failure2;
run;
```

A listing of FAILURE2 is shown in Figure 34.2.

Obs	cause	COUNT	PERCENT
1	Contamination	14	45.1613
2	Corrosion	2	6.4516
3	Doping	1	3.2258
4	Metallization	2	6.4516
5	Miscellaneous	3	9.6774
6	Oxide Defect	8	25.8065
7	Silicon Defect	1	3.2258

Figure 34.2. The Data Set FAILURE2 Created Using PROC FREQ

The following statements produce a Pareto chart for the data in FAILURE2:

```
title 'Analysis of IC Failures';
proc pareto data=failure2;
   hbar cause / freq       = count
                scale      = count
                interbar   = 1.0
                last       = 'Miscellaneous'
                nlegend    = 'Total Circuits'
                cframenleg = empty ;
run;
```

The chart is displayed in Figure 34.3.

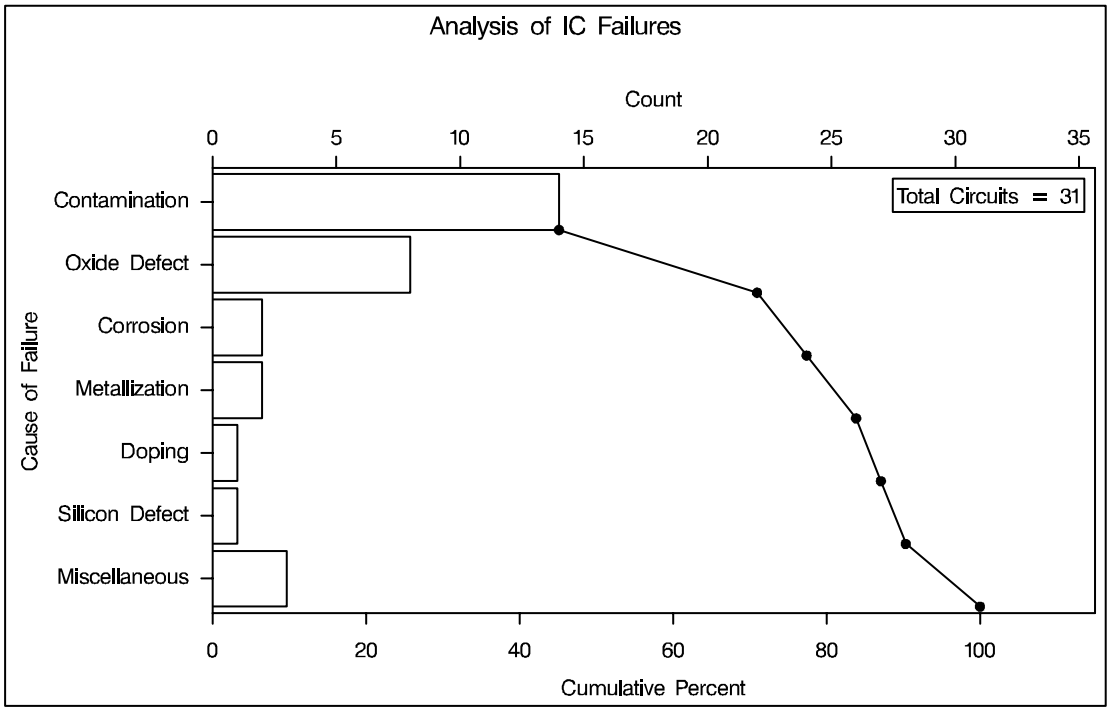


Figure 34.3. Pareto Chart with Frequency Scale

A slash (/) is used to separate the process variable CAUSE from the options specified in the HBAR statement. The frequency variable COUNT is specified with the FREQ= option. Specifying the keyword COUNT with the SCALE= option requests a frequency scale for the horizontal axis.

The INTERBAR= option inserts a small space between the bars, and specifying LAST='Miscellaneous' causes the category *Miscellaneous* to be displayed last regardless of its frequency. The NLEGEND= option adds a sample size legend labeled *Total Circuits*, and the CFRAMENLEG= option frames the legend. The SYMBOL statement marks points on the curve with dots.

There are two sets of tied categories in this example; *Corrosion* and *Metallization* each occur twice, and *Doping* and *Silicon Defect* each occur once. The procedure displays tied categories in alphabetical order of their formatted values. Thus, *Corrosion* appears before *Metallization*, and *Doping* appears before *Silicon Defect* in Figure 34.3. This is simply a convention, and no practical significance should be attached to the order in which tied categories are arranged.

Restricting the Number of Pareto Categories

Unlike the previous examples, some applications involve too many categories to display on a chart. The solution presented here is to create a restricted Pareto chart that displays only the most frequently occurring categories.

The following statements create a Pareto chart for the five most frequently occurring levels of CAUSE in the data set FAILURE2, which is listed in Figure 34.2:

```
title 'IC Manufacturing Problems';
proc pareto data=failure2;
   hbar cause / freq     = count
                scale    = count
                maxncat  = 5;
run;
```

The MAXNCAT= option specifies the number of categories to be displayed. The chart, shown in Figure 34.4, does not display the categories *Doping* and *Silicon Defect*.

You can also display the most frequently occurring categories and merge the remaining categories into a single *other* category that is represented by a bar. You can specify the name for the new category with the OTHER= option. If, in addition, you specify the name with the LAST= option, the category is positioned at the bottom of the chart. The following statements illustrate both options:

```
title 'IC Manufacturing Problems';
proc pareto data=failure2;
   hbar cause / freq       = count
                scale      = count
                maxncat    = 5
                other      = 'Others'
                last       = 'Others';
run;
```

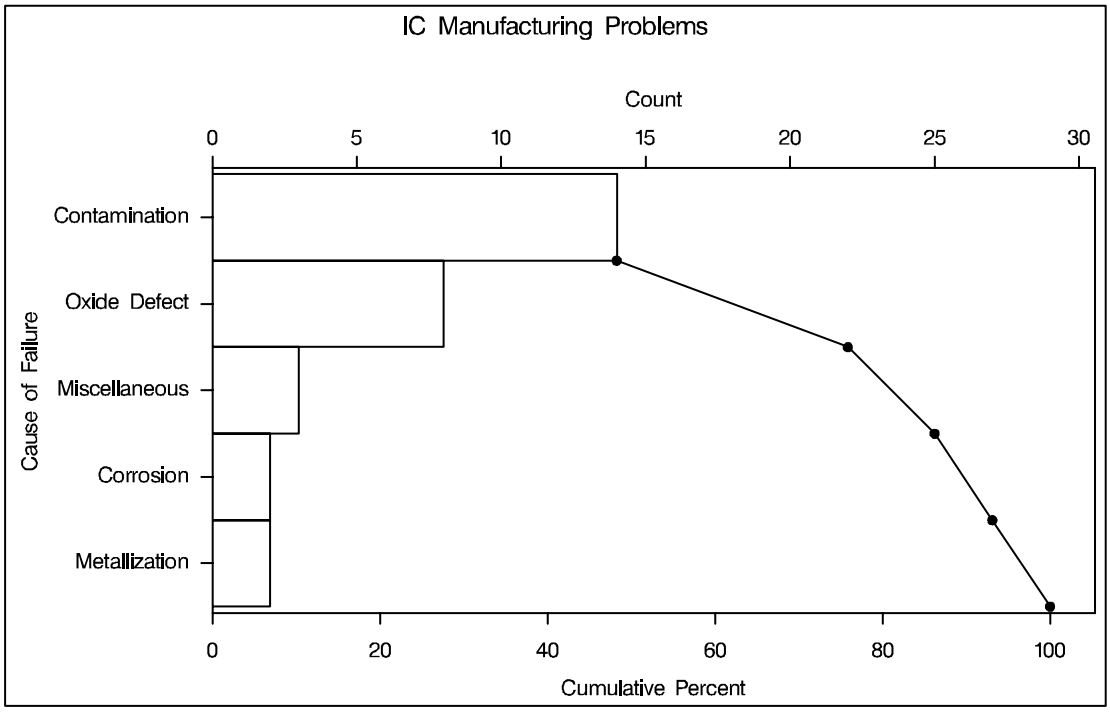


Figure 34.4. Restricted Pareto Chart

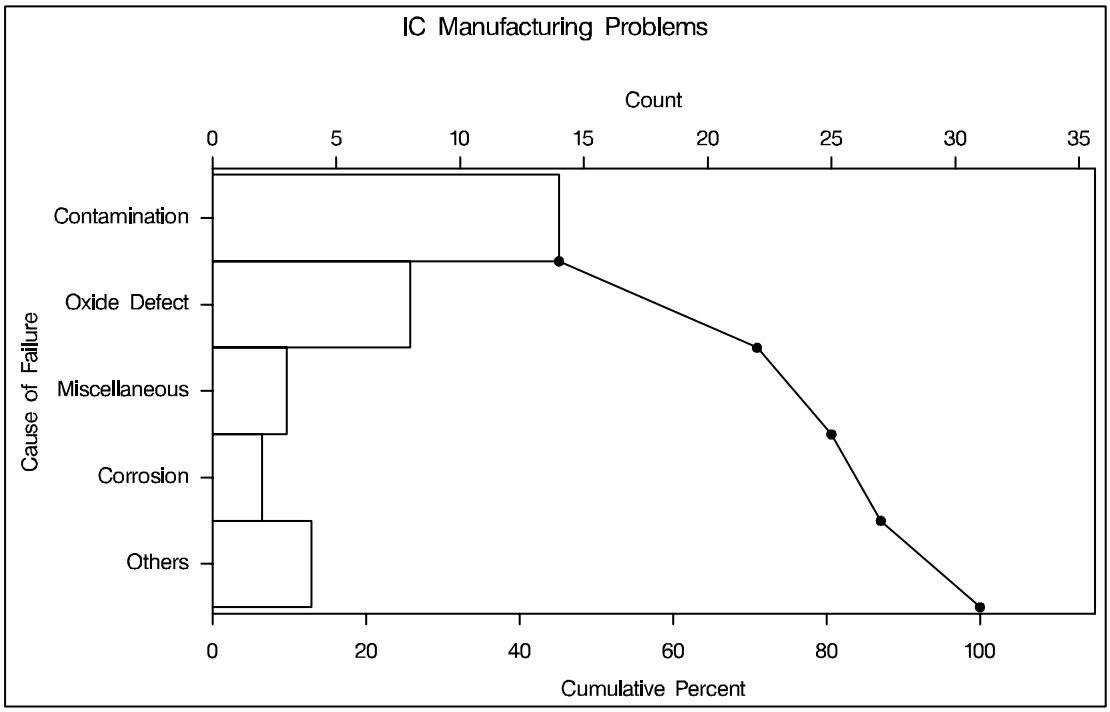


Figure 34.5. Restricted Pareto Chart with *Other* Category

The chart is shown in Figure 34.5.

The number of categories displayed is five, which is the number specified with the MAXNCAT= option. The first four categories are the four most frequently occurring problems in FAILURE2, and the fifth category merges the remaining problems.

Note that *Corrosion* and *Metallization* both have a frequency of two. When the MAXNCAT= option is applied to categories with tied frequencies, the procedure breaks the tie by using the order of the formatted values. Thus *Corrosion* is displayed, whereas *Metallization* is merged into the *Other* category. The MAXNCAT= and related options are described in "Restricted Pareto Charts" on page 1050.

Syntax

The syntax for the HBAR statement is as follows:

HBAR (*variable-list***) < / *options* > ;**

You can use any number of HBAR statements in the PARETO procedure. If you specify two or more *variables* in the HBAR statement, they must be enclosed in parentheses. The components of the HBAR statement are described as follows.

options
specify the layout and features of the chart, and they are listed after a slash (/) that follows the variables to be analyzed.

The "Summary of Options" section, which follows, provides summary tables of options organized by function. The "Dictionary of Options" on page 1010 describes the *options* in detail.

variable-list
specifies the process variables to be analyzed. A chart is created for each *variable*, and the values of each *variable* determine the Pareto categories for that chart. A list of two or more *variables* must be enclosed in parentheses.

The *variables* can be numeric or character, and the maximum length of a character *variable* is 32. Formatted values are used to determine the categories and are displayed in labels and legends. The maximum format length is 32.

Summary of Options

The following tables list the HBAR statement options by function. For complete descriptions, see "Dictionary of Options" on page 1010.

Table 34.1. Options for the Cumulative Percent Curve

ANCHOR=*keyword*	specifies corner of topmost bar to which curve is anchored
CCONNECT=*color*	specifies color for curve
CMPCTLABEL	labels curve points with their values
NOCURVE	suppresses curve
NOHLABEL2	suppresses secondary horizontal axis label
NOHTICK2	suppresses secondary horizontal axis tick marks and tick mark labels

Table 34.2. Data Processing Options

CFRAMENLEG=*color*	frames the NLEGEND legend and fills the frame with the specified color
FREQ=*variable*	specifies frequency variable
MISSING	specifies that missing values of the process variable be treated as a Pareto category
MISSING1	specifies that missing values of the first CLASS= variable be analyzed as a level
MISSING2	specifies that missing values of the second CLASS= variable be analyzed as a level
NLEGEND	requests sample size legend
NLEGEND='*label*' | (*variable*)	requests sample size legend with specified label
OUT=*SAS-data-set*	creates output data set that saves information displayed in the Pareto chart
WEIGHT=*variable-list*	specifies weight variables used to weight frequencies

Table 34.3. Options for Restricting the Number of Categories

COTHER=*color*	specifies color for OTHER= bar
LOTHER='*label*'	specifies label for OTHER= bar
MAXCMPCT=*percent*	displays only the categories with cumulative percentage less than the *percent* specified
MAXNCAT=n	displays only the categories with the n highest values
MINPCT=*percent*	displays only the categories with percents greater than the *percent* specified
OTHER='*category*'	merges all categories not displayed
OTHERCVAL='*value*'	specifies an OUT= data set character variable value for the OTHER= category
OTHERNVAL=*value*	specifies an OUT= data set numeric variable value for the OTHER= category
POTHER=*pattern*	specifies pattern for OTHER= bar

Table 34.4. Options to Enhance Plots Produced on Graphics Devices

ANNOTATE= *SAS-data-set*	specifies annotate data set with primary horizontal axis data units
ANNOTATE2= *SAS-data-set*	specifies annotate data set with secondary horizontal axis data units
CTEXT=*color*	specifies color for text
CTEXTSIDE=*color*	specifies color for row labels
CTEXTTOP=*color*	specifies color for column labels
DESCRIPTION='*string*'	specifies description for graphics catalog member
FONT=*font*	specifies font for text
HEIGHT=*value*	specifies text height in percent screen units
HTML=(*variable*)	specifies a variable whose values are URLs to be associated with bars
INFONT=*font*	specifies font for text inside frame
INHEIGHT=*value*	specifies text height in percent screen units for text inside frame
NAME='*string*'	specifies name for graphics catalog member

Table 34.5. Options for Reference Lines

CHREF=*color*	specifies color for HREF= and HREF2= lines
CVREF=*color*	specifies color for VREF= lines
FRONTREF	draws reference lines in front of bars
HREF=*value-list*	requests reference lines perpendicular to primary horizontal axis
HREF2=*value-list*	requests reference lines perpendicular to secondary horizontal axis
HREFLABELS= ('*label1*'...'*labeln*')	specifies labels for HREF= lines
HREF2LABELS= ('*label1*'...'*labeln*')	specifies labels for HREF2= lines
HREFLABPOS=*n*	specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=*linetype*	specifies line type for HREF= and HREF2= lines
LVREF=*linetype*	specifies line type for VREF= lines
VREF=*value-list*	requests reference lines perpendicular to vertical axis
VREFLABELS= ('*label1*'...'*labeln*')	specifies labels for VREF= lines
VREFLABPOS=*n*	specifies position of VREFLABELS= labels

Table 34.6. Options for Comparative Pareto Charts

ANNOKEY	applies annotation only to the key cell
CFRAMESIDE=*color*	specifies frame color for row labels
CFRAMETOP=*color*	specifies frame color for column labels
CLASS=(*variable-list*)	specifies classification variables
CLASSKEY= (*'value1' 'value2'*)	specifies the key cell
CPROP=*color*	specifies color for proportion-of-frequency bar
CTILES=(*variable*)	specifies colors for tile backgrounds
INTERTILE=*value*	specifies distance in percent screen units between tiles
MISSING1	specifies that missing values of the first CLASS= variable be analyzed as a level
MISSING2	specifies that missing values of the second CLASS= variable be analyzed as a level
NCOLS=*n*	specifies number of columns
NOKEYMOVE	suppresses the placement of the key cell in the top left corner
NROWS=*n*	specifies number of rows
ORDER1=*keyword*	specifies the order in which values of the first CLASS= variable are displayed
ORDER2=*keyword*	specifies the order in which values of the second CLASS= variable are displayed
TILELEGEND=(*variable*)	specifies legend for CTILES= colors
TILELEGLABEL=*'label'*	specifies label for TILELEGEND= legend

Table 34.7. Options for Grids

CGRID=*color*	specifies color for GRID lines
CGRID2=*color*	specifies color for GRID2 lines
GRID	adds grid corresponding to primary horizontal axis
GRID2	adds grid corresponding to secondary horizontal axis
LGRID=*linetype*	specifies line type for GRID lines
LGRID2=*linetype*	specifies line type for GRID2 lines
WGRID=*n*	specifies width of GRID lines
WGRID2=*n*	specifies width of GRID2 lines

Table 34.8. Options for Controlling Axes

ANGLE=*value*	rotates vertical axis tick mark labels
AXISFACTOR=*value*	specifies distance factor between the longest bar and the right frame
CAXIS=*color*	specifies axis color
CAXIS2=*color*	specifies color for secondary horizontal axis and tick marks
CFRAME=*color*	specifies color for area enclosed by axes and frame
HAXIS=*value-list*	specifies tick mark values for primary horizontal axis
HAXISLABEL=*'label'*	labels primary horizontal axis
HAXIS2=*value-list*	specifies tick mark values for secondary horizontal axis
HAXIS2LABEL=*'label'*	labels secondary horizontal axis
HOFFSET=*value*	specifies horizontal axis offset in percent screen units
NOCHART	suppresses Pareto chart
NOFRAME	suppresses axis frame
NOHLABEL	suppresses primary horizontal axis label
NOHLABEL2	suppresses secondary horizontal axis label
NOHTICK	suppresses tick marks and tick mark labels for primary horizontal axis
NOHTICK2	suppresses tick marks and tick mark labels for secondary horizontal axis
NOVLABEL	suppresses vertical axis label
SCALE=*keyword*	specifies units in which primary horizontal axis is scaled
VOFFSET=*value*	specifies vertical axis offset in percent screen units
WAXIS=n	specifies width in pixels for the axes and frame

Table 34.9. Options for Displaying a Sample Size Legend

CFRAMENLEG=*color*	frames the NLEGEND legend and fills the frame with the specified color
NLEGEND	requests sample size legend
NLEGEND='*label*'|(*variable*)	requests sample size legend with specified label

Table 34.10. Options for Displaying Bars

BARLABEL=*keyword*|(*variable-list*)	displays labels for bars
BARLABPOS=*keyword*	specifies position of BARLABEL= option
BARLEGEND=(*variable-list*)	displays legend for CBARS= colors or PBARS= patterns
BARLEGLABEL='*label*'	displays label for BARLEGEND= legend
BARWIDTH=*value*	specifies width (vertical dimension) in percent screen units of the bars
CATLEGLABEL='*label*'	specifies label for Pareto categories legend
CBARLINE=*color*	specifies color for bar outlines
CBARS=*color*	specifies color for bars
CBARS=*variable-list*	specifies variable that provides bar colors
CHIGH(n)=*color*	specifies color for bars with the n highest values
CLOW(n)=*color*	specifies color for bars with the n lowest values
HLLEGLABEL='*label*'	displays label for the legend that describes colors and patterns of highest or lowest bars
INTERBAR=*value*	specifies distance between bars in percent screen units
LABOTHER='*other-label*'	specifies label for "other" category
LAST='*category*'	specifies bottommost category
NOHLLEG	suppresses legend describing colors and patterns of highest or lowest bars
PBARS=*pattern*	specifies pattern for the bars
PBARS=(*variable-list*)	specifies variable that provides bar patterns
PHIGH(n)=*pattern*	specifies pattern for bars with the n highest values
PLOW(n)=*pattern*	specifies pattern for bars with the n lowest values
WBARLINE=*n*	specifies width for bar outlines

Dictionary of Options

The following entries provide detailed descriptions of options you can specify after the slash (/) in the HBAR statement. For example, to request that the bars of your Pareto chart be colored red, use the CBARS= option, as follows:

```
proc pareto data=failure;
   hbar cause / cbars = red ;
run;
```

ANCHOR=BR | LC | RC | TL

specifies where the Pareto curve is anchored to the topmost bar on the chart. The following table lists the possible positions.

Keyword	Anchoring position
BR	bottom right corner
LC	left center
RC	right center
TL	top left corner

See Output 36.2.1 on page 1062 for an illustration. The default is BR.

ANGLE=*value*
specifies an angle in degrees for rotating the Pareto category labels on the vertical axis. The *value* is the angle between the baseline of the label and the vertical axis. See Output 36.1.1 on page 1057 and Output 36.1.2 on page 1057 for an illustration. The *value* must be greater than -90 and less than 90. If you specify the ANGLE= option, you should also specify a software font with the FONT= option in the HBAR statement or the FTEXT= option in a GOPTIONS statement. The default *value* is zero.

ANNOKEY
specifies that annotation requested with the ANNOTATE= and ANNOTATE2= options is to be applied only to the key cell in a comparative Pareto chart. By default, annotation is applied to all of the cells.

ANNOTATE=*SAS-data-set*
ANNO=*SAS-data-set*
specifies an input data set that contains annotate variables as described in *SAS/GRAPH Software: Reference*. You can use this data set to customize charts. This data set is associated with the primary horizontal axis. If the annotation is based on data coordinates, you must use the same units as the primary horizontal axis. Features provided in the ANNOTATE= data set are added to every chart produced with the HBAR statement.

ANNOTATE2=*SAS-data-set*
ANNO2=*SAS-data-set*
specifies an input data set that contains annotate variables as described in *SAS/GRAPH Software: Reference*. You can use this data set to customize charts. This data set is associated with the secondary horizontal axis. If the annotation is based on data coordinates, you must use the same units as the secondary horizontal axis. Features provided in the ANNOTATE2= data set are added to every chart produced with the HBAR statement.

AXISFACTOR=*value*
specifies a factor used in scaling the primary horizontal axis. This factor determines (approximately) the ratio of the length of the axis to the length of the longest bar, and it is used to provide space for the cumulative percent curve. The *value* must be greater than one.

By default, the factor is chosen so that the curve will be anchored at the bottom right corner of the topmost bar (see also the ANCHOR= option). If this causes the bars to be flattened excessively, however, a smaller default factor is used.

The AXISFACTOR= option is not applicable if the curve is suppressed with the NOCURVE option.

BARLABEL = VALUE | CMPCT | (*variable*)
BARLABEL=VALUE | CMPCT
specifies that a label is to be displayed to the right of each bar. If you specify BARLABEL=VALUE, the label indicates the length of the bar in the units used by the primary horizontal axis. See Example 36.8 on page 1075 for an illustration.

If you specify BARLABEL=CMPCT, the label indicates the cumulative percent for that bar. An alternative to BARLABEL=CMPCT is the CMPCTLABEL option, which labels points on the cumulative percent curve with the cumulative percents.

If you specify BARLABEL= (*variable*), the label indicates the values of the variable specified in parentheses. The variable can have a formatted length less than or equal to 32. If a format is associated with the variable, then the formatted value is displayed. The values must be consistent within observations corresponding to a particular Pareto category. The variable is saved in the OUT= data set.

BARLABPOS=HBAR | HFIT | HRIGHT
specifies the position for labels requested with the BARLABEL= option.

- BARLABPOS=HBAR displays the labels on the bars.
- BARLABPOS=HFIT displays the labels on or to the right of the bars, depending on the available space.
- BARLABPOS=HRIGHT displays the labels to the right of the bars.

The labels are truncated if necessary. By default, the labels are displayed to the right of the bars.

BARLEGEND=(*variable-list*)
specifies that a legend is to be added to the chart to explain colors for bars specified with the CBARS=(*variable-list*) option or patterns for bars specified with the PBARS=(*variable-list*) option. The *variable-list* must be enclosed in parentheses even if only one *variable* is specified. See Output 36.4.1 on page 1069 for an illustration.

The values of the BARLEGEND= variable provide the explanatory labels used in the legend. The variable can have a formatted length that does not exceed 32. If a format is associated with the variable, then the formatted value is displayed.

The BARLEGEND= option is not applicable unless you specify CBARS=(*variable-list*) or PBARS=(*variable-list*) or both. In the DATA= data set, the values of the BARLEGEND= variable must be identical in observations for which the value of the CBARS= variable or the PBARS= variable (or the combination of these two values) is the same. This ensures that the legend derived from the BARLEGEND= variable is consistent.

If you specify more than one process variable in the chart statement and a corresponding list of CBARS= or PBARS= variables, you can specify a list of BARLEGEND=

variables. The number of BARLEGEND= variables should be less than or equal to the number of process variables. The lists of variables are matched so that the first BARLEGEND= variable is applied to the first process variable and the first CBARS= or PBARS= variable, the second BARLEGEND= variable is applied to the second process variable and the second CBARS= or PBARS= variable, and so forth. If the list of process variables is longer than the *variable-list* in the BARLEGEND= option, the charts for the extra process variables will not display a bar legend.

BARLEGLABEL=*'label'*
specifies the label displayed to the left of the legend created with the BARLEGEND= option. See Output 36.4.1 on page 1069 for an illustration.

The BARLEGLABEL= option is applicable only in conjunction with CBARS= or PBARS= variables. The label can be up to 16 characters and must be enclosed in quotes.

If no label is specified with the BARLEGLABEL= option, the label associated with the BARLEGEND= variable is displayed (unless the label is longer than 16 characters, in which case the variable name is displayed). If the BARLEGLABEL= option is not specified and no label is associated with the BARLEGEND= variable, no legend label is displayed. If both labels are specified, the BARLEGLABEL= label takes precedence over the variable label.

BARWIDTH=*value*
specifies the *width* of the bars in percent screen units. The width of a bar on a horizontal Pareto chart refers to its vertical dimension. By default, the bars are as wide as possible.

CATLEGLABEL=*'label'*
specifies a label for the category legend that is added when there is insufficient space to label the categories along the vertical axis. The *label* can be up to 16 characters and must be enclosed in quotes. The default label is *Categories:*. See Example 36.3 on page 1066 for an illustration. The CATLEGLABEL= option is ignored if it is unnecessary to add the legend.

CAXIS=*color*
CAXES=*color*
CA=*color*
specifies the color for the axis line and tick marks on the chart. The default color is the first color in the device color list. This color is also used for bar outlines and grid lines, unless overridden by the CBARLINE=, CGRID=, or CGRID2= options.

CAXIS2=*color*
specifies the color for the tick mark labels and axis label associated with the secondary axis. By default, the color specified with the CTEXT= option (or its default) is used.

CBARLINE=*color*
specifies the color for bar outlines. By default, bar outlines are the same color as the axes.

CBARS=*color*
CBARS=(*variable-list***)**
specifies how the bars of the Pareto chart are to be colored. You can use one of the following approaches:

- You can specify a single color to be used for all the bars with CBARS=*color*. You can use this option in conjunction with the CHIGH and CLOW options. See Output 36.2.1 on page 1062 for an illustration.
- You can specify a distinct color for each bar (or combination of bars) by providing the colors as values of a CBARS= variable. This variable must be a character variable of length eight. You can use the special value `EMPTY` to indicate that a bar is not to be colored. Note that the variable name must be enclosed in parentheses. You cannot specify a CBARS= variable in conjunction with the CHIGH and CLOW options. See Output 36.3.1 on page 1067 and Output 36.4.1 on page 1069 for examples.

If you specify more than one process variable, you can specify more than one CBARS= variable. The number of CBARS= variables should be less than or equal to the number of process variables. The two lists of variables are paired in order of specification. If a CBARS= variable is not provided for a process variable, the bars for that chart are not colored.

If you specify one or more CBARS= variables, you can also use the BARLEGEND= option to add a legend to the chart that explains the significance of each color. Furthermore, you can use the PBARS= option to specify patterns in conjunction with the CBARS= option. See Output 36.4.1 on page 1069 and Output 36.5.1 on page 1070 for examples.

CCONNECT=*color*
specifies the color for the line segments connecting the points on the cumulative percent curve. You can specify the color for the points themselves with the COLOR= option in the SYMBOL statement; this is the default color.

CFRAME=*color*
specifies the color for filling the area enclosed by the axes and the frame. By default, this area is not filled. The CFRAME= option cannot be used in conjunction with the NOFRAME option or the CTILES= option.

CFRAMENLEG=*color*
specifies that the legend requested with the NLEGEND option be framed and that the frame be filled with the color indicated. If you specify CFRAMENLEG=EMPTY, a frame is drawn but is not filled with a color. See Figure 34.3 on page 1002 and Output 36.1.4 on page 1059 for illustrations.

CFRAMESIDE=*color*
specifies the color for filling the frame area for the row labels displayed along the left side of a comparative Pareto chart requested with the CLASS= option. If a label is associated with the classification variable, this color is also used to fill the frame area for this label. By default, these areas are not filled.

CFRAMETOP=*color*
specifies the color for filling the frame area for the column labels displayed across the top of a comparative Pareto chart requested with the CLASS= option. If a label is associated with the classification variable, this color is also used to fill the frame area for this label. By default, these areas are not filled.

CGRID=*color*
specifies the color for grid lines requested with the GRID option. By default, grid lines are the same color as the axes. If you specify the CGRID= option, you do not need to specify the GRID option.

CGRID2=*color*
specifies the color for grid lines requested with the GRID2 option. By default, grid lines are the same color as the axes. If you specify the CGRID2= option, you do not need to specify the GRID2 option.

CHIGH(n**)=***color*
specifies the color used to fill the bars with the n highest values. You cannot use the CHIGH option in conjunction with a CBARS= variable, but you can use the CHIGH(n)= option together with the CLOW(n)= and CBARS=*color* options. See Output 36.3.1 on page 1067 for an illustration. By default, the bars are empty.

CHREF=*color*
specifies the color for lines requested with the HREF= and HREF2= options. The default is the first color in the device color list.

CLASS=*variable*
CLASS=(*variable1 variable2***)**
creates a comparative Pareto chart using the levels of the *variables*. You must enclose two *variables* in parentheses. See Example 36.1 on page 1056 and Example 36.2 on page 1060.

If you specify a single CLASS= *variable*, the observations in the input data set are classified by the formatted values (levels) of the *variable*. A Pareto chart is created for the process variable values in each level, and these component charts (referred to as cells) are arranged in an array. The cells are labeled with the levels, and uniform horizontal and vertical axes are used to facilitate comparisons.

If you specify two CLASS= *variables*, the observations in the input data set are cross-classified by the values (levels) of the *variables*. A Pareto chart is created for the process variable values in each cell of the cross-classification, and these charts are arranged in a *matrix*. The levels of the first CLASS= *variable* label the rows, and the levels of the second CLASS= *variable* label the columns. Uniform horizontal and vertical axes are used to facilitate comparisons.

The CLASS= *variables* can be numeric or character. The maximum length of a character *variable* is 32. If a format is associated with a CLASS= *variable*, the formatted values determine the levels. Only the first 32 characters of the formatted value are used to determine the levels. You can specify whether missing values are to be treated as a level with the MISSING1 and MISSING2 options.

If a label is associated with a CLASS= *variable*, the label is displayed on the chart, parallel to the column (or row) labels.

CLASSKEY=*'value'*
CLASSKEY=(*'value1' 'value2'***)**
specifies the *key cell* in a comparative histogram requested with the CLASS= option. The bin size and midpoints are first determined for the key cell, and then the midpoint list is extended to accommodate the data ranges for the remaining cells. Thus the choice of the key cell determines the uniform vertical axis used for all cells.

If you specify CLASS=*variable*, you can specify CLASSKEY=*'value'* to identify the key cell as the level for which *variable* is equal to *value*. The *value* can have up to 32 characters, and you must specify a formatted *value*. By default, the levels are sorted in the order determined by the ORDER1= option, and the key cell is the level that occurs first in this order. The cells are displayed in this order from top to bottom (or left to right), and consequently the key cell is displayed at the top or at the left. If you specify a different key cell with the CLASSKEY= option, this cell is displayed at the top or at the left unless you also specify the NOKEYMOVE option.

If you specify CLASS=(*variable1 variable2*), you can specify CLASSKEY=(*'value1' 'value2'*) to identify the key cell as the level for which *variable1* is equal to *value1* and *variable2* is equal to *value2*. Here, *value1* and *value2* must be formatted values, and they must be enclosed in quotes. By default, the levels of *variable1* are sorted in the order determined by the ORDER1= option, and then within each of these levels, the levels of *variable2* are sorted in the order determined by the ORDER2= option. The default key cell is the combination of levels of *variable1* and *variable2* that occurs first in this order. The cells are displayed in order of *variable1* from top to bottom and in order of *variable2* from left to right. Consequently, the default key cell is displayed in the upper left corner. If you specify a different key cell with the CLASSKEY= option, this cell is displayed in the upper left corner unless you also specify the NOKEYMOVE option.

For an example of the use of the CLASSKEY= option, see Output 36.1.3 on page 1058.

CLOW(n**)=***color*
specifies the color used to fill the bars with the n lowest values. You cannot use the CLOW(n)= option in conjunction with a CBARS=*variable*, but you can use the CLOW(n)= option together with the CBARS=*color* and CHIGH options. See Output 36.3.1 on page 1067 for an illustration of the CHIGH(n)= option. By default, the bars are empty.

CMPCTLABEL
labels points on the cumulative percent curve with their values. By default, the points are not labeled.

COTHER=*color*
specifies the color for the bar defined by the OTHER= option. By default, this bar is not filled with a color. The COTHER= option is not applicable unless a CBARS=*variable* is specified.

CPROP=*color*

specifies the color for a proportion-of-frequency bar that is displayed vertically along the right side of each tile in a comparative Pareto chart. The length of the bar relative to the height of the tile indicates the proportion of the total frequency count in the chart that is represented by the tile. You can use the bars to visualize the distribution of frequency count by tile. See Output 36.1.4 on page 1059 for an illustration.

The CPROP= option provides a graphical alternative to the NLEGEND options, which display the actual count. The CPROP= option is applicable only with comparative Pareto charts. Empty bars are displayed if you specify CPROP=EMPTY. Bars are not displayed if the CPROP= option is not specified.

CTEXT=*color*
CT=*color*

specifies the color for text, such as tick mark labels, axis labels, and legends. The default is the value specified for the CTEXT= option in the GOPTIONS statement.

CTEXTSIDE=*color*

specifies the color for row labels displayed along the left side of a comparative Pareto chart requested with the CLASS= option. The default color is the color specified with the CTEXT= option in the HBAR statement or the CTEXT= option in the GOPTIONS statement.

CTEXTTOP=*color*

specifies the color for column labels displayed across the top of a comparative Pareto chart requested with the CLASS= option. The default color is the color specified with the CTEXT= option in the HBAR statement or the CTEXT= option in the GOPTIONS statement.

CTILES=(*variable***)**

specifies a character variable of length eight whose values are the fill colors for the tiles in a comparative Pareto chart. The CTILES= option generalizes the CFRAME= option, which provides a single color for all of the tiles. The *variable* must be enclosed in parentheses. The values of the *variable* must be identical for all observations with the same level of the CLASS= variables. You can use the same color to fill more than one tile. You can use the special value `EMPTY` to indicate that a tile is not to be filled.

The CTILES= option cannot be used in conjunction with the NOFRAME option or the CFRAME= option. You can use the TILELEGEND= option in conjunction with the CTILES= option to add an explanatory legend for the CTILES= colors at the bottom of the chart. See Output 36.5.1 on page 1070 for an illustration. By default, the tiles are not filled.

CVREF=*color*

specifies the color for lines requested with the VREF= lines. The default color is the first color in the device color list.

DESCRIPTION=*'string'*
DES=*'string'*
specifies a descriptive string, up to 40 characters, that appears in the description field of the PROC GREPLAY master menu.

FONT=*font*
specifies a software font for text used in labels and legends. The FONT= option takes precedence over the FTEXT= option in the GOPTIONS statement.

FREQ=*variable*
specifies a frequency variable whose value provides the counts (numbers of occurrences) of the values of the process variable. Specifying a FREQ= variable is equivalent to replicating the observations in the input data set. The FREQ= variable must be a numeric variable with nonnegative integer values. If you specify more than one process variable in the chart statement, the FREQ= variable values are used with each process variable. See "Creating a Pareto Chart Using Frequency Data" on page 1001 for an illustration. If you do not specify a FREQ= variable, each value of the process variable is counted exactly once.

FRONTREF
Graphics
draws reference lines requested with the HREF= and VREF= options in front of the bars on the Pareto chart. By default, reference lines are drawn behind the bars and can be obscured by them.

GRID
adds a grid to the Pareto chart corresponding to the primary horizontal axis. Grid lines are vertical lines positioned at tick marks on the primary horizontal axis. The lines are useful for comparing the lengths of the bars.

GRID2
adds a grid to the Pareto chart corresponding to the secondary horizontal axis. Grid lines are vertical lines positioned at tick marks on the secondary horizontal axis. The lines are useful for reading the cumulative percent curve.

HAXIS=*value-list*
specifies tick mark values for the primary horizontal axis. The values must be equally spaced and in increasing order, and the first *value* must be zero. You must scale the values in the same units as the bars (see the SCALE= option), and the last *value* must be greater than or equal to the height of the largest bar.

HAXISLABEL=*'label'*
specifies a label, up to 40 characters, for the primary horizontal axis. The default label depends on the value of the SCALE= option, or it is the label associated with the WEIGHT= variable.

HAXIS2=*value-list*
specifies tick mark values for the secondary horizontal axis. The values must be equally spaced and in increasing order, and the first *value* must be zero. You must scale the values in percent units, and the last *value* must be greater than or equal to 100.

HAXIS2LABEL=*'label'*
specifies a label, up to 40 characters, for the secondary horizontal axis. The default label is *Cumulative Percent* or *Cm Pct*, depending on the space available.

HEIGHT=*value*
specifies the height in percent screen units of text for labels and legends. This option should be used only in conjunction with the FONT= option. The HEIGHT= option takes precedence over the HTEXT= option in a GOPTIONS statement.

HLLEGLABEL=*'label'*
specifies a label displayed to the left of the legend that is automatically created when you use a combination of the CHIGH, CLOW, PHIGH, and PLOW options. See Output 36.3.1 on page 1067 for an illustration. The *label* can be up to 16 characters and must be enclosed in quotes. The default *label* is *Bars:*.

HOFFSET=*value*
specifies the length in percent screen units of the offset at the right end of the primary horizontal axis. You can eliminate the offset by specifying HOFFSET=0.

HREF=*value-list*
specifies where reference lines perpendicular to the primary horizontal axis are to appear on the chart. You must specify the values in the same units used to scale the primary axis. By default, the primary axis is scaled in percent units, but you can specify other units with the SCALE= option. See Output 36.2.3 on page 1064 for an illustration.

HREF2=*value-list*
specifies where reference lines perpendicular to the secondary horizontal axis are to appear on the chart. You must specify the values in cumulative percent units.

HREFLABELS=*'label1'...'labeln'*
specifies labels for the lines requested with the HREF= option. The number of labels must equal the number of lines requested. Labels can be up to 16 characters and must be enclosed in quotes.

HREF2LABELS=*'label1'...'labeln'*
specifies labels for the lines requested with the HREF2= option. The number of labels must equal the number of lines requested. Labels can be up to 16 characters and must be enclosed in quotes.

HREFLABPOS=*n*
specifies the vertical positioning of the HREFLABELS= labels. HREFLABPOS=1 positions the labels along the top of the chart. HREFLABPOS=2 staggers the labels from top to bottom. HREFLABPOS=3 positions the labels along the bottom. By default, HREFLABPOS=1.

HTML=*variable*
specifies URLs as values of the specified character variable (or formatted values of a numeric variable). These URLs are associated with bars on the Pareto chart when graphics output is directed into HTML. The value of the HTML= variable should be the same for each observation with a given value of the subgroup variable.

INFONT=*font*
specifies a software font for text used inside the frame of the chart, such as sample size legends. The INFONT= option takes precedence over the FONT= option and the FTEXT= option in the GOPTIONS statement.

INHEIGHT=*value*
specifies the height in percent screen units of text used inside the frame of the chart, such as sample size legends and bar labels. This option should be used in conjunction with the INFONT= option.

INTERBAR=*value*
specifies the distance in percent screen units between bars on the chart. By default, the bars are contiguous. See Figure 34.3 on page 1002 for an illustration.

INTERTILE=*value*
specifies the distance in horizontal percent screen units between tiles (cells) in a comparative Pareto chart. By default, the tiles are contiguous. See Output 36.1.3 on page 1058 for an illustration.

LABOTHER = *'other-label'*
is used in conjunction with the BARLABEL=(variable) option and specifies a label for the "other" category that is optionally specified with the OTHER= option.

LAST=*'category'*
specifies that the bar corresponding to the *category* is to be displayed at the bottom of the chart regardless of the percent associated with this category. The *category* must be a formatted value of the process variable and must be enclosed in quotes. The *category* can be up to 32 characters. See Figure 34.5 on page 1004 for an illustration.

LGRID=*line-type*
specifies the line type for the grid requested with the GRID option. The default *line-type* is 1, which produces a solid line. If you specify the LGRID= option, you do not need to specify the GRID option.

LGRID2=*line-type*
specifies the line type for the grid requested with the GRID2 option. The default *line-type* is 1, which produces a solid line. If you specify the LGRID2= option, you do not need to specify the GRID2 option.

LHREF=*line-type*
LH=*line-type*
specifies the line type for lines requested with the HREF= and HREF2= options. The default *line-type* is 2, which produces a dashed line.

LOTHER=*'label'*
specifies a label for the bar defined with the OTHER= option. This label appears in the legend created with the BARLEGEND= option. The *label* must be enclosed in quotes and can be up to 32 characters. The default *label* is the value specified with the OTHER= option. The LOTHER= option is applicable only when a BARLEGEND= variable is specified.

LVREF=*line-type*
LV=*line-type*
specifies the line type for lines requested with the VREF= option. See Output 36.2.3 on page 1064 for an illustration. The default *line-type* is 2, which produces a dashed line.

MAXCMPCT=*percent*
specifies that only the Pareto categories with the n highest frequency counts are to be displayed, where the sum of the n corresponding percents is less than or equal to the specified *percent*. For example, if you specify

```
proc pareto data=failure;
   hbar cause / maxcmpct = 90 ;
```

the chart displays only the n most frequently occurring categories that account for no more than 90 percent of the total frequency.

You can use the OTHER= option in conjunction with the MAXCMPCT= option to create and display a new category that combines those categories that are not selected with the MAXCMPCT= option. For example, if you specify

```
proc pareto data=failure;
   hbar cause / maxcmpct = 90
                other    = 'Others' ;
```

the chart displays the categories that account for no more than 90 percent of the total frequency, together with a category labeled *Others* that merges the remaining categories. The MAXCMPCT= option is an alternative to the MINPCT= and MAXNCAT= options.

MAXNCAT=*n*
specifies that only the Pareto categories with the n highest frequencies are to be displayed. For example, if you specify

```
proc pareto data=failure;
   hbar cause / maxncat = 20 ;
```

the chart displays only the categories with the 20 highest frequencies. If the total number of categories is less than 20, all the categories are displayed.

You can use the OTHER= option in conjunction with the MAXNCAT= option to create and display a new category that combines those categories that are not selected with the MAXNCAT= option. For example, if you specify

```
proc pareto data=failure;
   hbar cause / maxncat = 20
                other= 'Others' ;
```

the chart displays the categories with the 19 highest frequencies, together with a category labeled *Others* that merges the remaining categories. See Figure 34.4 on page 1004 for another illustration.

The MAXNCAT= option is an alternative to the MINPCT= and MAXCMPCT= options.

MINPCT=*percent*

specifies that only the Pareto categories with frequency percents greater than or equal to the specified *percent* are to be displayed. For example, if you specify

```
proc pareto data=failure;
   hbar cause / minpct = 5 ;
```

the chart displays only those categories with at least five percent of the total frequency.

You can use the OTHER= option in conjunction with the MINPCT= option to create and display a new category that combines those categories that are not selected with the MINPCT= option. The merged category created by the OTHER= option is displayed even if its total percent is less than the *percent* specified with the MINPCT= option. For example, if you specify

```
proc pareto data=failure;
   hbar cause / minpct = 5
                other  = 'Others' ;
```

the chart displays the categories with percents greater than or equal to five percent, together with a category labeled *Others* that merges the remaining categories.

The MINPCT= option is an alternative to the MAXNCAT= and MAXCMPCT= options.

MISSING

specifies that missing values of the process variable are to be treated as a Pareto category represented with a bar on the chart. If the process variable is a character variable, a missing value is defined as a blank internal (unformatted) value. If the process variable is numeric, a missing value is defined as any of the SAS missing values. If you do not specify the MISSING option, missing values are excluded from the analysis.

MISSING1

specifies that missing values of the first CLASS= variable are to be treated as a level of the CLASS= variable. If the first CLASS= variable is a character variable, a missing value is defined as a blank internal (unformatted) value. If the first CLASS= variable is numeric, a missing value is defined as any of the SAS missing values. If you do not specify MISSING1, observations in the DATA= data set for which the first CLASS= variable is missing are excluded from the analysis.

MISSING2
specifies that missing values of the second CLASS= variable are to be treated as a level of the CLASS= variable. If the second CLASS= variable is a character variable, a missing value is defined as a blank internal (unformatted) value. If the second CLASS= variable is numeric, a missing value is defined as any of the SAS missing values. If you do not specify MISSING2, observations in the DATA= data set for which the second CLASS= variable is missing are excluded from the analysis.

NAME=*'string'*
specifies a name for the chart, up to eight characters, that appears in the PROC GREPLAY master menu. The default name is 'PARETO'.

NCOLS=*n*
NCOL=*n*
specifies the number of columns in a comparative Pareto chart. You can use the NCOLS= option in conjunction with the NROWS= option. See Output 36.2.3 (page 1064) and Output 36.2.4 (page 1065) for an illustration. By default, NCOLS=1 and NROWS=2 if one CLASS= variable is specified, and NCOLS=2 and NROWS=2 if two CLASS= variables are specified.

NLEGEND
NLEGEND='*label*'
NLEGEND=(*variable***)**
requests a sample size legend and specifies its form as follows:

- If you specify the NLEGEND option, the form is *N*=*n*, where n is the total count for the Pareto categories. In a comparative Pareto chart, a legend is displayed in each tile, and n is the total count for that particular cell. See Output 36.2.1 on page 1062 for an illustration.
- If you specify the NLEGEND='*label*' option, the form is *label*=*n*, where n is the total count for the Pareto categories. The label can be up to 32 characters and must be enclosed in quotes. For an illustration, see Figure 34.3 on page 1002 or Output 36.1.4 on page 1059.
- If you specify the NLEGEND=(*variable*) option, the legend is the value of the *variable*, which must be a variable in the DATA= data set whose formatted length does not exceed 32. If a format is associated with the variable, then the formatted value is displayed. This option is intended for use with comparative Pareto charts and enables you to display a customized legend inside each tile (this legend need not provide total count). It is assumed that the values of the *variable* are identical for all observations in a particular class.

By default, the legend is placed in the upper-right corner of the chart. If the NOCURVE option is specified, the legend is placed in the lower-right corner of the chart. You can use the CFRAMENLEG= option to frame the sample size legend. No legend is displayed if you do not specify an NLEGEND option.

NOCHART
suppresses the creation of a Pareto chart. This option is useful when you are simply creating an output data set.

NOCURVE
suppresses the display of the cumulative percent curve and the secondary horizontal axis. Compare Output 36.2.1 (page 1062) and Output 36.2.2 (page 1063) for an illustration.

NOFRAME
suppresses the frame that is drawn around the chart by default. The NOFRAME option cannot be specified in conjunction with the CFRAME= or CTILES= options.

NOHLABEL
suppresses the label for the primary horizontal axis.

NOHLABEL2
suppresses the label for the secondary horizontal axis. This is useful for avoiding clutter on comparative Pareto charts.

NOHLLEG
suppresses the legend generated by the CHIGH(n)=, CLOW(n)=, PHIGH(n)=, and PLOW(n)= options.

NOHTICK
suppresses the primary horizontal axis label, tick marks, and tick mark labels.

NOHTICK2
suppresses the secondary horizontal axis label, tick marks, and tick mark labels.

NOKEYMOVE
suppresses the rearrangement of cells within a comparative Pareto chart that occurs when you use the CLASSKEY= option. The key cell appears in the top left corner of a comparative Pareto chart unless you use the CLASSKEY= option together with the NOKEYMOVE option.

NOVLABEL
suppresses the label for the vertical axis. This is useful for avoiding clutter in situations where the meaning of the vertical axis is apparent from the labels for the Pareto categories. See Output 36.2.2 on page 1063 for an illustration.

NROWS=n
NROW=n
specifies the number of rows in a comparative Pareto chart. You can use the NROWS= option in conjunction with the NCOLS= option. See Output 36.2.3 on page 1064 and Output 36.2.4 on page 1065 for an illustration. By default, NROWS=2.

ORDER1=INTERNAL | FORMATTED | DATA | FREQ
specifies the display order for the values of the first CLASS= variable. The levels of the first CLASS= variable are always constructed using the formatted values of the variable, and the formatted values are always used to label the rows (columns) of a comparative Pareto chart.

If you specify ORDER1=INTERNAL, the rows (columns) are displayed from top to bottom (left to right) in increasing order of the internal, or unformatted, values of the first CLASS= variable. If there are two or more distinct internal values with the same formatted value, the order is determined by the internal value that occurs first in the input data set. For example, suppose that you use a numeric CLASS= variable called DAY (with values 1, 2, and 3) to create a one-way comparative Pareto chart. Suppose also that you use the FORMAT procedure to associate the formatted values 1 = 'Wednesday', 2 = 'Thursday', and 3 = 'Friday' with the variable DAY. If you specify ORDER1=INTERNAL, the rows of the comparative chart will appear in chronological order (*Wednesday*, *Thursday*, *Friday*) from top to bottom.

If you specify ORDER1=FORMATTED, the rows (columns) are displayed from top to bottom (left to right) in increasing order of the formatted values of the first CLASS= variable. For instance, in the previous illustration, if you specify ORDER1=FORMATTED, the rows will appear in alphabetical order (*Friday*, *Thursday*, *Wednesday*) from top to bottom.

If you specify ORDER1=DATA, the rows (columns) are displayed from top to bottom (left to right) in the order in which the values of the first CLASS= variable first appear in the input data set.

If you specify ORDER1=FREQ, the rows (columns) are displayed from top to bottom (left to right) in order of *decreasing* frequency count. If two or more classes have the same frequency count, the order is determined by the formatted values.

By default, ORDER1=INTERNAL.

ORDER2=INTERNAL | FORMATTED | DATA | FREQ

specifies the display order for the values of the second CLASS= variable. The levels of the second CLASS= variable are always constructed using the formatted values of the variable, and the formatted values are always used to label the columns of a two-way comparative Pareto chart.

The PARETO procedure determines the layout of a two-way comparative Pareto chart by first using the ORDER1= option to obtain the order of the rows from top to bottom (recall that ORDER1=INTERNAL by default). Then the ORDER2= option is applied to the observations corresponding to the first row to obtain the order of the columns from left to right. If any columns remain unordered (that is, the categories are unbalanced), the ORDER2= option is applied to the observations in the second row, and so on until all the columns have been ordered.

The values of the ORDER2= option are interpreted as described for the ORDER1= option. By default, ORDER2=INTERNAL.

OTHER=*'category'*

specifies a new category that merges all categories not selected with the MAXNCAT=, MINPCT=, or MAXCMPCT= options. See Figure 34.5 on page 1004 for an illustration.

The *category* should be specified as a formatted value of the process variable. The *category* can be up to 32 characters and must be enclosed in quotes. If you specify

an OUT= data set, you should also specify an internal value corresponding to the *category* with the OTHERCVAL= option or the OTHERNVAL= option.

The OTHER= option is not applicable unless you specify the MAXNCAT=, MINPCT=, or MAXCMPCT= option. You can use the COTHER=, LOTHER=, POTHER=, OTHERCVAL=, and OTHERNVAL= options with the OTHER= option.

OTHERCVAL=*'value'*
specifies the internal (unformatted) value for a character process variable in the OUT= data set that corresponds to the category created with the OTHER= option. The *category* can be up to 32 characters and must be enclosed in quotes.

The OTHERCVAL= option is not applicable unless you specify the OTHER= and OUT= options. If you specify the OTHER= option but not the OTHERCVAL= option, the default *value* is the *value* specified with the OTHER= option.

OTHERNVAL=*value*
specifies the internal (unformatted) value for a numeric process variable in the OUT= data set that corresponds to the category created with the OTHER= option. The OTHERNVAL= option is not applicable unless you specify the OTHER= and OUT= options. If you specify the OTHER= option but not the OTHERNVAL= option, *value* is assigned a missing value.

OUT=*SAS-data-set*
creates an output data set that contains the information displayed in the Pareto chart. This is useful if you want to create a report to accompany your chart. See Example 36.8 on page 1075 for an illustration.

PBARS=*pattern*
PBARS=(*variable-list***)**
specifies pattern fills for the bars. You can use one of two approaches:

- You can specify a single pattern to be used for all the bars with the PBARS=*pattern* option. You can use this option in conjunction with the PHIGH and PLOW options. See Output 36.2.1 on page 1062 for an illustration.
- You can specify a distinct pattern for *each* bar (or combination of bars) by providing the patterns as values of a PBARS= variable. For example, you might use the solid pattern (`S`) to indicate severe problems and the empty pattern (`E`) for all other problems. The variable must be a character variable of length eight, and the variable name must be enclosed in parentheses. You cannot specify a PBARS= variable in conjunction with the PHIGH and PLOW options. See Output 36.4.1 on page 1069 and Output 36.5.1 on page 1070 for illustrations.

If you specify more than one process variable in the chart statement, you can provide more than one PBARS= variable. The number of PBARS= variables should be less than or equal to the number of process variables. The two lists of variables are paired in order of specification. If a PBARS= variable is not provided for a process variable, the bars for that chart are not filled.

If you specify one or more variables with the PBARS= option, you can also use the BARLEGEND= option to add a legend to the chart that explains the significance

of each pattern. Furthermore, you can use the CBARS= option to specify colors in conjunction with the PBARS= option. See Output 36.4.1 on page 1069 and Output 36.5.1 on page 1070 for illustrations.

PHIGH(*n*)=*pattern*

specifies the pattern used to fill the bars with the n highest values. You cannot specify the PHIGH option in conjunction with a PBARS= variable, but you can specify the PHIGH(n)= option together with the PLOW(n)= and PBARS=*pattern* options. See Output 36.3.1 on page 1067 for an illustration. By default, the bars are empty.

PLOW(*n*)=*pattern*

specifies the pattern used to fill the bars with the n lowest values. You cannot specify the PLOW option in conjunction with a PBARS= variable, but you can use the PLOW(n)= option together with the PHIGH(n)= and PBARS=*pattern* options. See Output 36.3.1 on page 1067 for an illustration of the PHIGH(n)= option. By default, the bars are empty.

POTHER=*pattern*

specifies the pattern used for the bar defined by the OTHER= option. By default, this bar is empty. The POTHER= option is not applicable unless a PBARS= variable is specified.

SCALE=PERCENT | COUNT | WEIGHT

specifies the scale for the primary horizontal axis.

If you specify SCALE=PERCENT, the scale is percent of total frequency. If a WEIGHT= variable is used, the scale is percent of total weight.

If you specify SCALE=COUNT, the scale is counts. See Output 36.1.4 on page 1059 for an illustration. This option is not applicable if a WEIGHT= variable is used. You can specify SCALE=FREQUENCY instead of SCALE=COUNT.

If you specify SCALE=WEIGHT, the vertical axis is scaled in the same units as the WEIGHT= variable. This option is not applicable unless you use a WEIGHT= variable.

By default, SCALE=PERCENT. See Output 36.5.1 on page 1070 for an example. Regardless of how SCALE= is specified, the secondary axis is scaled in cumulative percent units.

TILELEGEND=(*variable*)

specifies a variable used to add a legend for CTILES= colors. The variable can have a formatted length less than or equal to 32. If a format is associated with the variable, then the formatted value is displayed. The TILELEGEND= option must be used in conjunction with the CTILES= option for filling the tiles in a comparative Pareto chart. If CTILES= is specified and TILELEGEND= is not specified, a color legend is not displayed.

The values of the CTILES= and TILELEGEND= variables should be consistent for all observations with the same level of the CLASS= variables. The value of the TILELEGEND= variable is used to identify the corresponding color value of the CTILES= variable in the legend. See Output 36.5.1 on page 1070 for an illustration.

TILELEGLABEL=*'label'*
specifies a label displayed to the left of the legend that is created when you specify a TILELEGEND= variable. The *label* can be up to 16 characters and must be enclosed in quotes. The default *label* is *Tiles:*. See Output 36.5.1 on page 1070 for an illustration.

VOFFSET=*value*
specifies the length in percent screen units of the offset at both ends of the vertical axis. You can eliminate the offset by specifying VOFFSET=0.

VREF=*'value-list'*
specifies where reference lines perpendicular to the vertical (Pareto category) axis are to appear on the chart. Character values can be up to 32 characters and must be enclosed in quotes. The values must be values of the process variable even when the bars are numbered and a category legend is introduced.

VREFLABELS=*'label1'...'labeln'*
specifies labels for the lines requested with the VREF= option. The number of labels must equal the number of lines requested. Enclose the labels in quotes. Labels can be up to 16 characters.

VREFLABPOS=*n*
specifies the vertical positioning of the VREFLABELS= labels. If you specify VREFLABPOS=1, the labels are positioned at the left of the chart, and if you specify VREFLABPOS=2, the labels are positioned at the right. By default, n=1.

WAXIS=*n*
specifies the line thickness (in pixels) for the axes and frame. By default, $n = 1$. This thickness is also used for bar outlines and grid lines, unless overridden by the WBARLINE=, WGRID=, or WGRID2= options.

WBARLINE=*n*
specifies the width for bar outlines. By default, bar outlines are the same width as the axes.

WEIGHT=*variable-list*
specifies weight variables used to construct weighted Pareto charts. The WEIGHT= variables are paired with the process variables in order of specification. The WEIGHT= variables must be numeric, and their values must be nonnegative (noninteger values are permitted). If a WEIGHT= variable is not provided for a process variable, the weights applied to that process variable are assumed to be one. See "Weighted Pareto Charts" on page 1050 for computational details.

A WEIGHT= variable is particularly useful for carrying out a Pareto analysis based on *cost* rather than frequency of occurrence. See Example 36.8 on page 1075 for an illustration.

WGRID=*n*
specifies the width of the primary chart grid lines. By default, grid lines are the same width as the axes. If the WGRID= option is specified the GRID option is not required.

WGRID2=*n*

specifies the width of the secondary chart grid lines. By default, grid lines are the same width as the axes. If the WGRID2= option is specified the GRID2 option is not required.

Chapter 35
INSET Statement

Chapter Contents

OVERVIEW 1033

GETTING STARTED 1033
- Displaying Summary Statistics on a Pareto Chart 1033
- Customizing Labels and Formatting Values 1035
- Adding a Header and Positioning the Inset 1037

SYNTAX 1038
- Summary of INSET Keywords 1040
- Summary of Options 1040
- Dictionary of Options 1041

DETAILS 1043
- Positioning the Inset Using Compass Points 1043
- Positioning the Inset in the Margins 1044
- Positioning the Inset Using Coordinates 1044

Chapter 35
INSET Statement

Overview

The INSET statement enables you to enhance a Pareto chart by adding a box or table (referred to as an *inset*) of summary statistics directly to the graph. An inset can display statistics calculated by the PARETO procedure or arbitrary values provided in a SAS data set.

Note that an INSET statement by itself does not produce a display but must be used in conjunction with a chart statement. Insets are not available with line printer output, so the INSET statement is not applicable when the LINEPRINTER option is specified in the PROC PARETO statement.

You can use options in the INSET statement to

- specify the position of the inset
- specify a header for the inset table
- specify graphical enhancements, such as background colors, text colors, text height, text font, and drop shadows

When an INSET statement is associated with a chart statement producing a comparative Pareto chart, an inset is produced in each cell of the comparative chart.

Getting Started

This section introduces the INSET statement with examples that illustrate commonly used options. Complete syntax for the INSET statement is presented in the "Syntax" section on page 1038.

Displaying Summary Statistics on a Pareto Chart

During the manufacture of a metal-oxide semiconductor (MOS) capacitor, causes of failures were recorded before and after a tube in the diffusion furnace was cleaned. This information was saved in a SAS data set named FAILURE3.

```
data failure3;
   length cause $ 16 stage $ 16 ;
   label  cause = 'Cause of Failure' ;
   input  stage $ 1-16 cause $ 19-34 counts;
datalines;
Before Cleaning   Contamination    14
Before Cleaning   Corrosion         2
Before Cleaning   Doping           1
```

```
Before Cleaning   Metallization     2
Before Cleaning   Miscellaneous     3
Before Cleaning   Oxide Defect      8
Before Cleaning   Silicon Defect    1
After Cleaning    Doping            0
After Cleaning    Corrosion         2
After Cleaning    Metallization     4
After Cleaning    Miscellaneous     2
After Cleaning    Oxide Defect      1
After Cleaning    Contamination    12
After Cleaning    Silicon Defect    2
run;
```

The following statements generate a Pareto chart from the FAILURE3 data. The MAXNCAT= option is specified to limit the number of Pareto categories to five. An INSET statement is used to display the total count for the categories displayed and the total count for the categories excluded by the MAXNCAT= option.

```
title 'IC Failures';
proc pareto data=failure3;
   vbar cause /
      freq     = counts
      maxncat  = 5;
   inset n nexcl / height = 3;
run;
```

The resulting chart is displayed in Figure 35.1. The INSET statement immediately follows the chart statement that creates the graphical display (in this case, the VBAR statement). Specify the keywords for inset statistics (such as N and NEXCL) immediately after the word INSET. The inset statistics appear in the order in which you specify the keywords. The HEIGHT= option in the INSET statement specifies the text height used to display the statistics in the inset.

A complete list of keywords that you can use with the INSET statement is provided in "Summary of INSET Keywords" on page 1040.

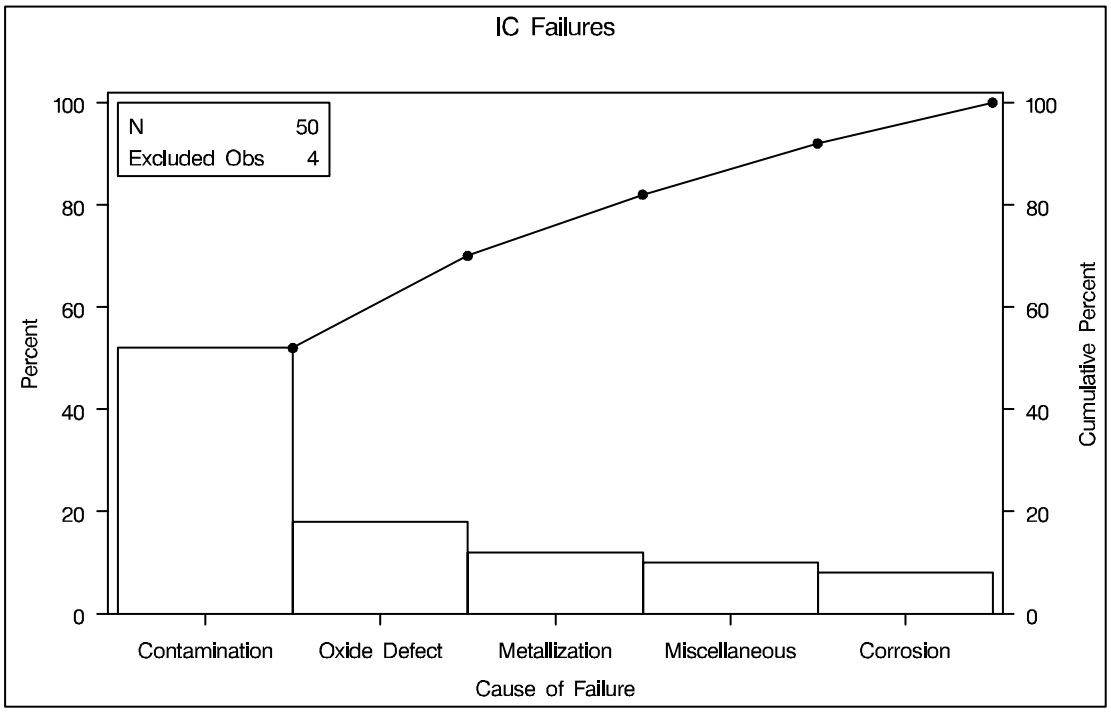


Figure 35.1. A Pareto Chart with an Inset

The following examples illustrate options commonly used for enhancing the appearance of an inset.

Customizing Labels and Formatting Values

By default, each inset statistic is identified with an appropriate label, and each numeric value is printed using an appropriate format. However, you may want to provide your own labels and formats. For example, in Figure 35.1 the default label used for the NEXCL statistic is rather long. The following statements produce a comparative Pareto chart with insets using a shorter label for the number of excluded observations. A format with one decimal place is also specified for each statistic. Note that a single INSET statement produces an inset in each cell of the comparative Pareto chart.

```
title 'Comparison of IC Failures';
proc pareto data=failure3;
   vbar cause /
      class    = stage
      freq     = counts
      maxncat  = 5
      classkey = 'Before Cleaning';
   inset n (3.1) nexcl='N Excl' (3.1) / height = 3;
run;
```

The resulting chart is displayed in Figure 35.2. You can provide your own label by specifying the keyword for that statistic followed by an equal sign (=) and the label in quotes. The label can have up to 24 characters.

The format 3.1 specified in parentheses after the N and NEXCL keywords displays those statistics with a field width of three and one decimal place. In general, you can specify any numeric SAS format in parentheses after an inset keyword. You can also specify a format to be used for all the statistics in the INSET statement with the FORMAT= option. For more information about SAS formats, refer to Chapter 14 of *SAS Language Reference: Dictionary*.

Note that if you specify both a label and a format for a statistic, the label must appear before the format.

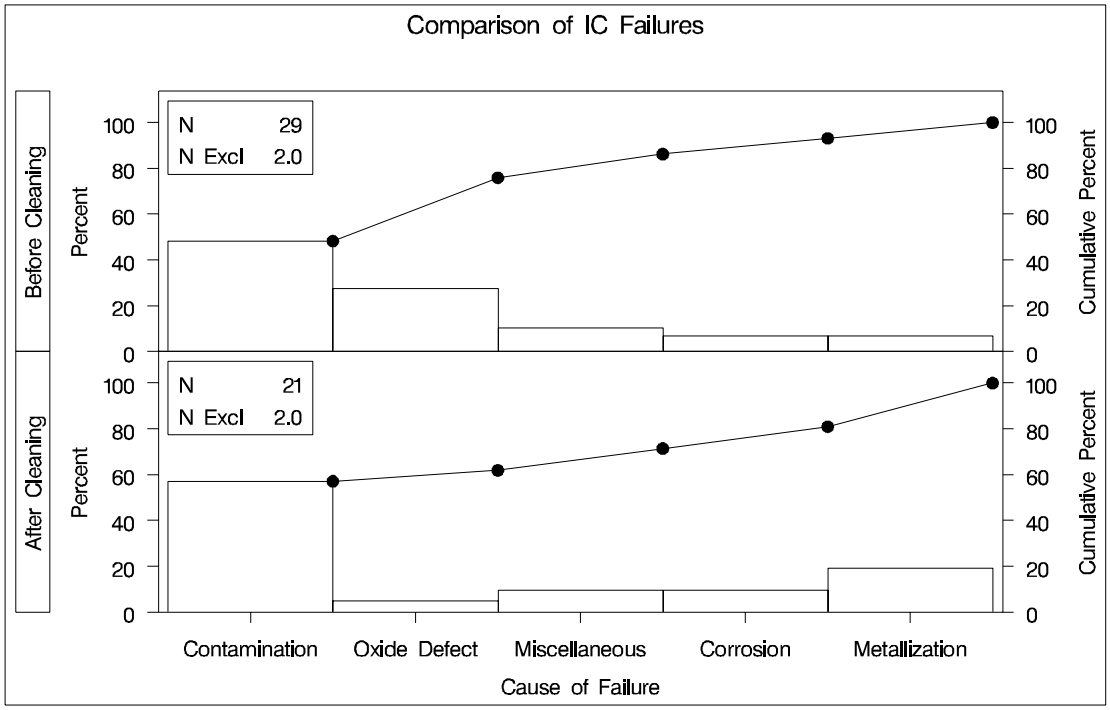


Figure 35.2. Customizing Labels and Formatting Values in an Inset

Adding a Header and Positioning the Inset

In the previous examples, the insets are displayed in the upper left corners of the plots, which is the default position for insets added to Pareto charts. You can control the inset position with the POSITION= option. In addition, you can display a header at the top of the inset with the HEADER= option. The following statements create a data set to be used with the INSET DATA= keyword and produce the horizontal Pareto chart shown in Figure 35.3:

```
data location;
   length _LABEL_ $ 10 _VALUE_ $ 12;
   input _LABEL_ _VALUE_ &;
datalines;
Plant      Santa Clara
Line       1
;


title 'IC Failures';
proc pareto data=failure3;
   hbar cause /
      freq     = counts
      maxncat  = 5;
   inset data = location n nexcl /
      height   = 3
      position = rm
      cshadow  = black
      header   = 'Count Summary';
run;
```

The header (in this case, *Count Summary*) can be up to 40 characters. Note that a longer list of inset statistics is requested. Consequently, POSITION=RM is specified to position the inset in the right margin so that it does not interfere with features of the chart. For more information about positioning, see "Details" on page 1043. The CSHADOW= option is used to display a drop shadow on this inset. The *options*, such as HEADER=, POSITION= and CSHADOW=, are specified after the slash (/) in the INSET statement. For more details on INSET statement options, see "Dictionary of Options" on page 1041.

Note that the contents of the data set LOCATION appear before other statistics in the inset. The position of the DATA= keyword in the keyword list determines the position of the data set's contents in the inset.

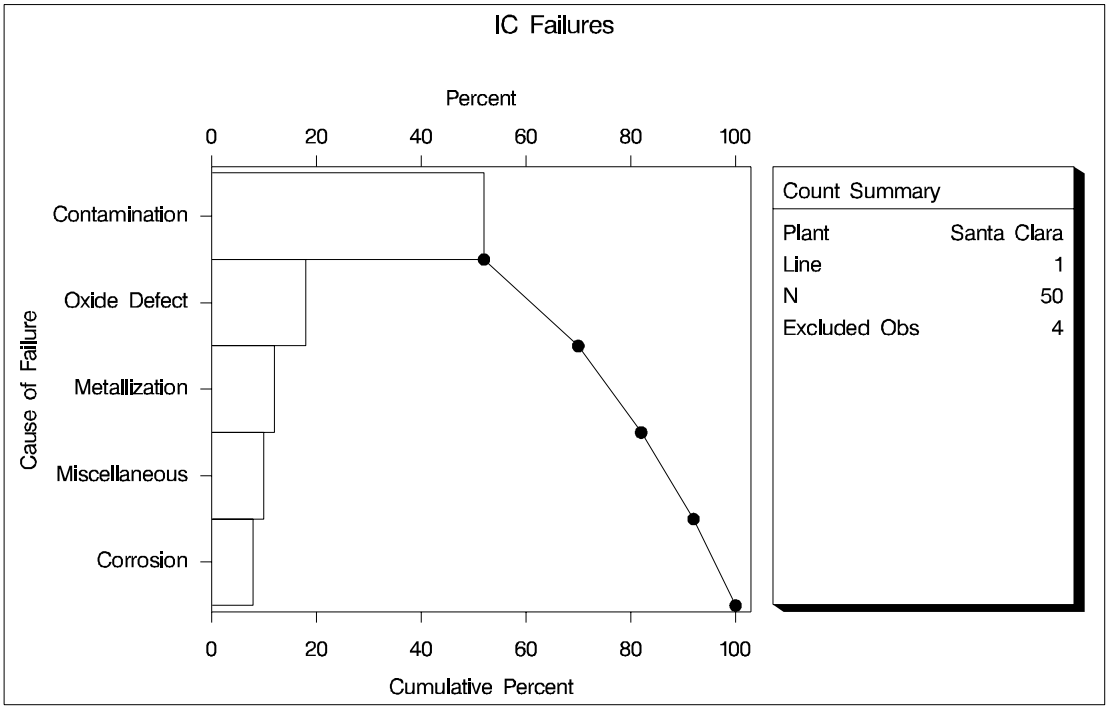


Figure 35.3. Adding a Header and Repositioning the Inset

Syntax

The syntax for the INSET statement is as follows:

INSET *keyword-list* < **/** *options* >**;**

You can use any number of INSET statements in the PARETO procedure. Each INSET statement produces a separate inset and must follow one of the chart statements. When the chart statement produces a comparative Pareto chart, an inset appears in every cell produced by the chart statement. The statistics are displayed in the order in which they are specified. The following statements produce a vertical Pareto chart with insets in the upper left and upper right corners, and a horizontal comparative Pareto chart with insets in each cell.

```
proc pareto data=failure3;
   vbar cause / maxncat = 5 other = 'Others';
      inset nothercat / pos = nw;
      inset nother / pos = ne;
   hbar cause / class = stage classkey = 'Before Cleaning';
      inset n / pos = ne;
run;
```

The statistics displayed in an inset are computed for a specific process variable using observations for the current BY group and CLASS= variable level, if applicable. For

example, in the following statements there are two process variables (TOMATO and SQUASH), a BY variable (YEAR), and two CLASS= variables (FERT and PEST). If there are three different years (levels of YEAR), then a total of six comparative Pareto charts are produced: three for each process variable. In addition, if there are two different levels of FERT and three of PEST, each comparative Pareto chart contains six cells. Each cell contains an inset with statistics computed for a particular process variable, year, and combination of FERT and PEST values.

```
proc pareto data=plants;
   by year;
   vbar (tomato squash) / class = (fert pest);
   inset n;
run;
```

The components of the INSET statement are described as follows.

keyword-list
can include any of the *keywords* listed in "Summary of INSET Keywords" on page 1040. The DATA= *keyword* requires an operand specified immediately after it, naming the data set containing data to be displayed.

The NOTHERCAT and NOTHER statistics are zero if the OTHER= option is not specified. The NEXCL statistic is zero if the OTHER= option *is* specified.

By default, inset statistics are identified with appropriate labels, and numeric values are printed using appropriate formats. However, you can provide customized labels and formats. You provide the customized label by specifying the *keyword* for that statistic followed by an equal sign (=) and the label in quotes. Labels can have up to 24 characters. You provide the numeric format in parentheses after the *keyword*. Note that if you specify both a label and a format for a statistic, the label must appear before the format. For an example, see "Customizing Labels and Formatting Values" on page 1035.

options
appear after the slash (/) and control the appearance of the inset. For example, the following INSET statement uses two appearance *options* (POSITION= and CTEXT=):

```
inset n nothercat nother / position=ne ctext=yellow;
```

The POSITION= option determines the location of the inset, and the CTEXT= option specifies the color of the text of the inset.

See "Summary of Options" on page 1040 for a list of all available *options* and "Dictionary of Options" on page 1041 for detailed descriptions. Note the difference between *keywords* and *options*; *keywords* specify the information to be displayed in an inset, whereas *options* control the appearance of the inset.

Summary of INSET Keywords

All keywords available with the PARETO procedure's INSET statement request a single statistic in an inset, except for the DATA= keyword.

The DATA= keyword specifies a SAS data set containing (label, value) pairs to be displayed in an inset. The data set must contain the variables _LABEL_ and _VALUE_. _LABEL_ is a character variable whose values provide labels for inset entries. _VALUE_ can be character or numeric, and it provides values displayed in the inset. The label and value from each observation in the DATA= data set occupy one line in the inset. Figure 35.3 shows an inset containing entries from a DATA= data set.

Table 35.1. Summary Statistics

N	sample size
NOTHERCAT	number of categories merged to form OTHER= category in restricted Pareto chart
NOTHER	number of observations in OTHER= category
NEXCL	observations excluded in restricted Pareto chart
SUMWGTS	sum across all categories of weighted frequencies
DATA=	(label, value) pairs from *SAS-data-set*

Summary of Options

The following table lists the INSET statement options. For complete descriptions, see "Dictionary of Options," which follows this section.

Table 35.2. INSET Options

CFILL=*color* | BLANK	specifies color of inset background
CFILLH=*color*	specifies color of header background
CFRAME=*color*	specifies color of frame
CHEADER=*color*	specifies color of header text
CSHADOW=*color*	specifies color of drop shadow
CTEXT=*color*	specifies color of inset text
DATA	specifies data units for POSITION=(x, y) coordinates
FONT=*font*	specifies font of text
FORMAT=*format*	specifies format of values in inset
HEADER='*quoted string*'	specifies header text
HEIGHT=*value*	specifies height of inset text
NOFRAME	suppresses frame around inset
POSITION=*position*	specifies position of inset
REFPOINT=BR|BL|TR|TL	specifies reference point of inset positioned with POSITION=(x, y) coordinates

Dictionary of Options

The following entries provide detailed descriptions of options for the INSET statement. Terms used in this section are illustrated in Figure 35.4.

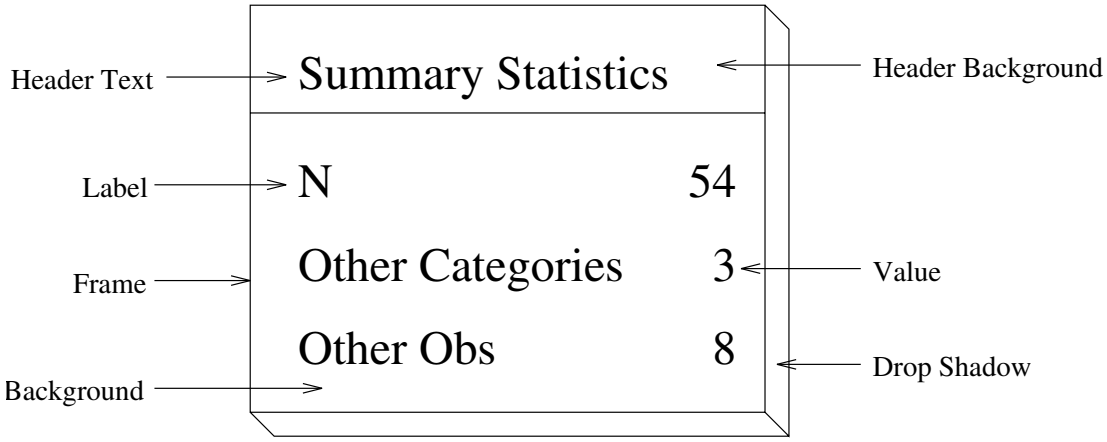


Figure 35.4. The Inset

CFILL=*color* **| BLANK**
specifies the color of the background (including the header background if you do not specify the CFILLH= option).

If you do not specify the CFILL= option, then, by default, the background is empty. This means that items that overlap the inset (such as subgroup data points or control limits) show through the inset. If you specify any value for the CFILL= option, then overlapping items no longer show through the inset. Specify CFILL=BLANK to leave the background uncolored and also to prevent items from showing through the inset.

CFILLH=*color*
specifies the color of the header background. By default, if you do not specify a CFILLH= color, the CFILL= color is used.

CFRAME=*color*
specifies the color of the frame. By default, the frame is the same color as the axis of the plot.

CHEADER=*color*
specifies the color of the header text. By default, if you do not specify a CHEADER= color, the CTEXT= color is used.

CSHADOW=*color*
CS=*color*
specifies the color of the drop shadow. See Figure 35.3 on page 1038 for an example. By default, if you do not specify the CSHADOW= option, a drop shadow is not displayed.

CTEXT=*color*
CT=*color*
specifies the color of the text. By default, the inset text color is the same as the other text on the plot.

DATA
specifies that data coordinates are to be used in positioning the inset with the POSITION= option. The DATA option is available only when you specify POSITION= (x, y), and it must be placed immediately after the coordinates (x, y). For details, see the entry for the POSITION= option or "Positioning the Inset Using Coordinates" on page 1044. See Figure 35.7 on page 1045 for an example.

FONT=*font*
specifies the font of the text. By default, the font is SIMPLEX if the inset is located in the interior of the plot, and the font is the same as the other text displayed on the plot if the inset is located in the exterior of the plot.

FORMAT=*format*
specifies a format for all the values displayed in an inset. If you specify a format for a particular statistic, then this format overrides the format you specified with the FORMAT= option.

HEADER= *'string'*
specifies the header text. The *string* cannot exceed 40 characters. If you do not specify the HEADER= option, no header line appears in the inset.

HEIGHT=*value*
specifies the height of the text.

NOFRAME
suppresses the frame drawn around the text.

POSITION=*position*
POS=*position*
determines the position of the inset. The *position* can be a compass point keyword, a margin keyword, or a pair of coordinates (x, y). You can specify coordinates in axis percent units or axis data units. For more information, see "Details" on page 1043. By default, POSITION=NW, which positions the inset in the upper left (northwest) corner of the display.

REFPOINT=BR | BL | TR | TL
RP=BR | BL | TR | TL
specifies the reference point for an inset that is positioned by a pair of coordinates with the POSITION= option. Use the REFPOINT= option with POSITION= coordinates. The REFPOINT= option specifies which corner of the inset frame you want positioned at coordinates (x, y). The keywords BL, BR, TL, and TR represent bottom left, bottom right, top left, and top right, respectively. See Figure 35.8 on page 1046 for an example. The default is REFPOINT=BL.

If you specify the position of the inset as a compass point or margin keyword, the REFPOINT= option is ignored. For more information, see "Positioning the Inset Using Coordinates" on page 1044.

Details

This section provides details on three different methods of positioning the inset using the POSITION= option. With the POSITION= option, you can specify

- compass points
- keywords for margin positions
- coordinates in data units or percent axis units

Positioning the Inset Using Compass Points

You can specify the eight compass points N, NE, E, SE, S, SW, W, and NW as keywords for the POSITION= option. The following statements create the display in Figure 35.5, which demonstrates all eight compass positions. The default is NW.

```
proc pareto data=failure3;
   vbar cause / freq = counts;
   inset n / height=3 cfill=blank header='NW' pos=nw;
   inset n / height=3 cfill=blank header='N ' pos=n ;
   inset n / height=3 cfill=blank header='NE' pos=ne;
   inset n / height=3 cfill=blank header='E ' pos=e ;
   inset n / height=3 cfill=blank header='SE' pos=se;
   inset n / height=3 cfill=blank header='S ' pos=s ;
   inset n / height=3 cfill=blank header='SW' pos=sw;
   inset n / height=3 cfill=blank header='W ' pos=w ;
run;
```

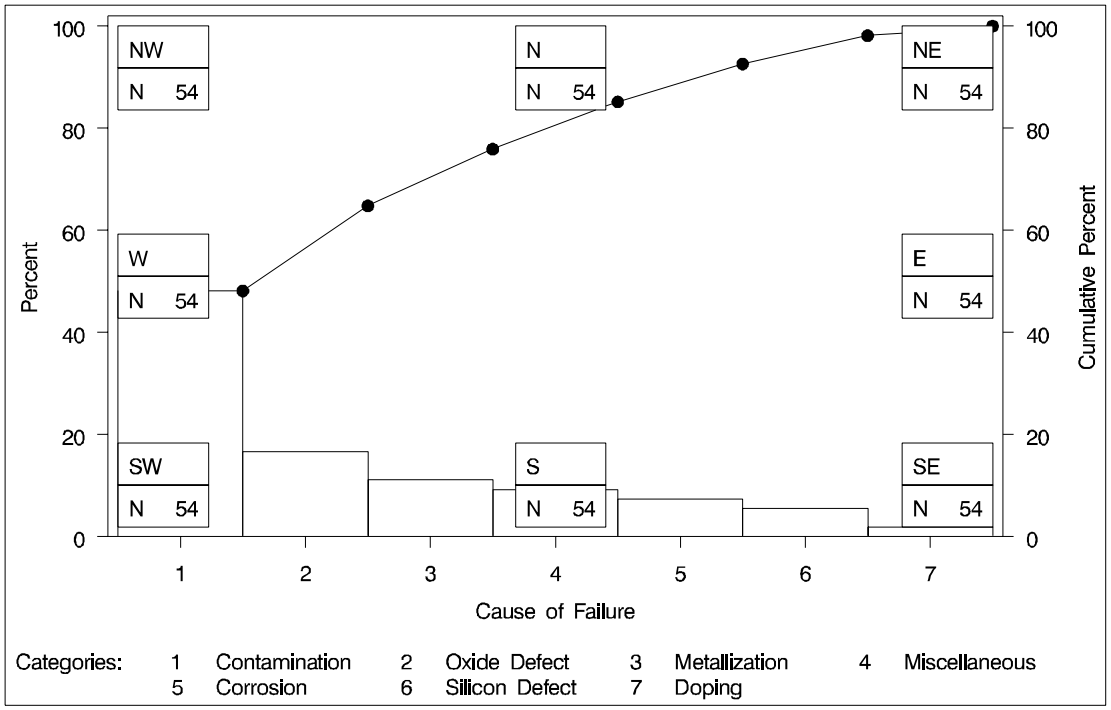


Figure 35.5. Insets Positioned Using Compass Points

Positioning the Inset in the Margins

Using the INSET statement you can also position an inset in one of the four margins surrounding the plot area using the margin keyword LM, RM, TM, or BM, as illustrated in Figure 35.6.

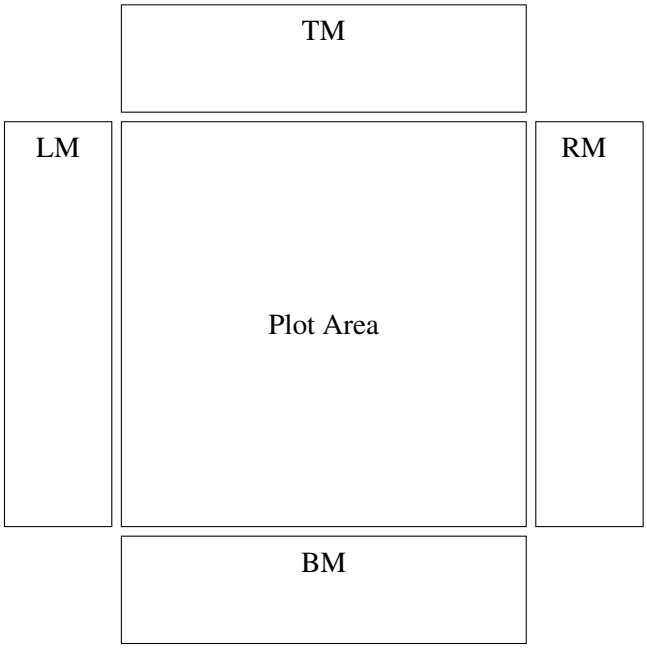


Figure 35.6. Positioning Insets in the Margins

For an example of an inset placed in the right margin, see Figure 35.3 on page 1038. Margin positions are recommended if a large number of statistics are listed in the INSET statement. If you attempt to display a lengthy inset in the interior of the plot, it is likely that the inset will collide with the data display.

Insets associated with a comparative Pareto chart cannot be positioned in the margins.

Positioning the Inset Using Coordinates

You can also specify the position of the inset with coordinates: POSITION= (x, y). The coordinates can be given in axis percent units (the default) or in axis data units.

Data Unit Coordinates

If you specify the DATA option immediately following the coordinates, the inset is positioned using axis data units. Data units along the category axis (the horizontal axis on a vertical bar chart or the vertical axis on a horizontal bar chart) are based on category numbers. Categories are numbered from left to right (vertical bar chart) or top to bottom (horizontal bar chart), starting with 1.

For example, the following statements produce the Pareto chart displayed in Figure 35.7:

```
proc pareto data=failure3;
   vbar cause / freq = counts;
   inset n / header   = 'Position=(3,60)'
             position = (3,60) data
             height   = 3;
run;
```

The bottom left corner of the inset is lined up with the tick mark for the third category on the horizontal axis and at 60 on the vertical axis. By default, the specified coordinates determine the position of the bottom left corner of the inset. You can change this reference point with the REFPOINT= option, as in the next example.

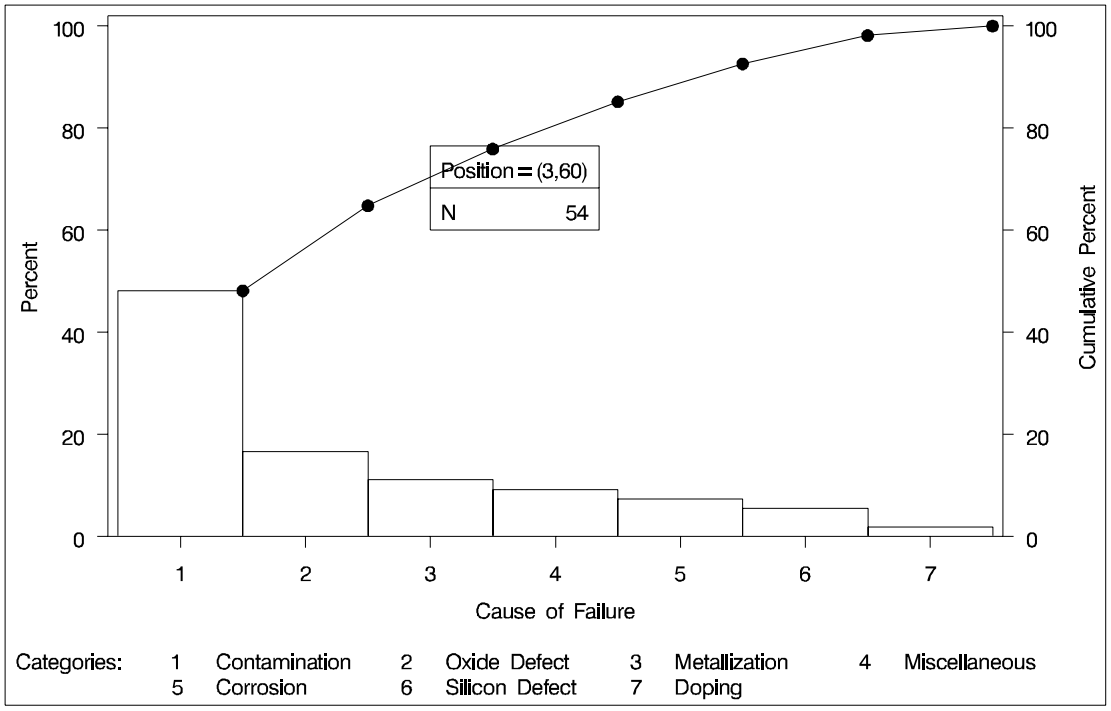


Figure 35.7. Inset Positioned Using Data Unit Coordinates

Axis Percent Unit Coordinates

If you do not use the DATA option, the inset is positioned using axis percent units. The coordinates of the bottom left corner of the display are $(0, 0)$, while the upper right corner is $(100, 100)$. For example, the following statements create a horizontal Pareto chart with two insets, both positioned using coordinates in axis percent units.

```
proc pareto data=failure3;
   vbar cause / freq     = counts
                maxncat  = 5;
   inset n / position = (5,25)
             header   = 'Position=(5,25)'
             height   = 3
             cfill    = blank
             refpoint = tl;
```

```
   inset nexcl / position = (95,95)
                 header   = 'Position=(95,95)'
                 height   = 3
                 cfill    = blank
                 refpoint = tr;
run;
```

The display is shown in Figure 35.8. Notice that the REFPOINT= option is used to determine which corner of the inset is to be placed at the coordinates specified with the POSITION= option. The first inset has REFPOINT=TL, so the top left corner of the inset is positioned 5% of the way across the horizontal axis and 25% of the way up the vertical axis. The second inset has REFPOINT=TR, so the top right corner of the inset is positioned 95% of the way across the horizontal axis and 95% of the way up the vertical axis. Note also that coordinates in axis percent units must be *between* 0 and 100.

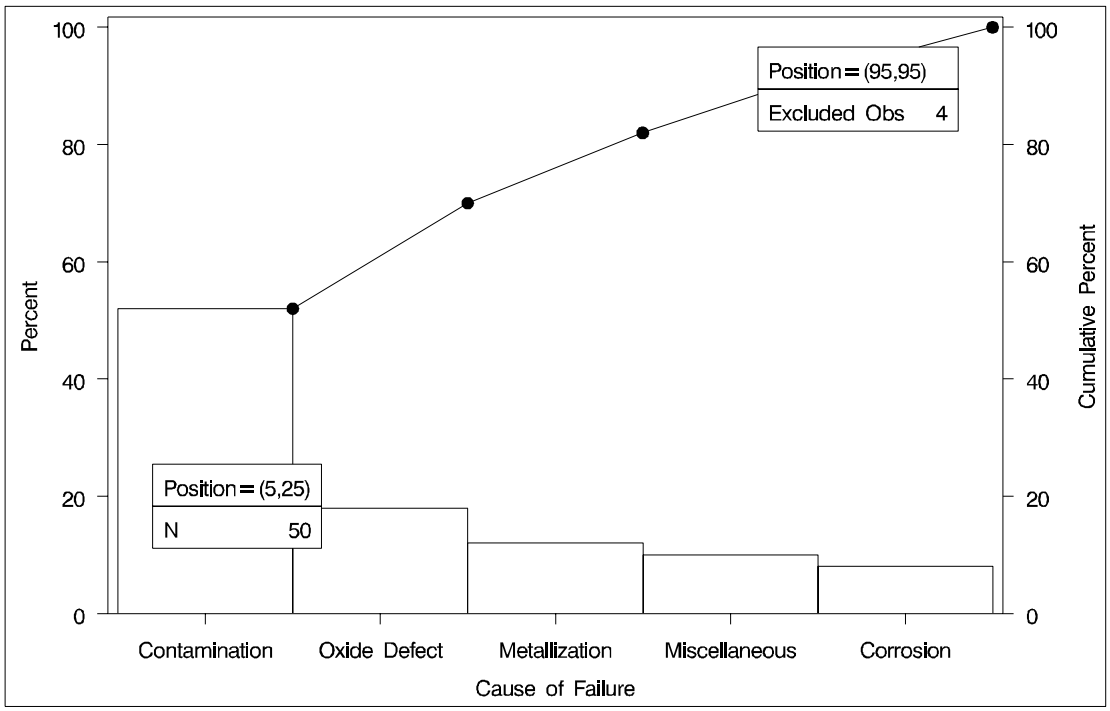


Figure 35.8. Inset Positioned Using Axis Percent Unit Coordinates

Chapter 36
Details and Examples

Chapter Contents

DETAILS 1049
Terminology 1049
Labels for Chart Features 1052
Scaling the Cumulative Percent Curve 1052
Output Data Sets 1053
Constructing Effective Pareto Charts 1054
Missing Values 1055
Role of Variable Formats 1055
Large Data Sets 1055

EXAMPLES 1056
Example 36.1. Creating Before-and-After Pareto Charts 1056
Example 36.2. Creating Two-Way Comparative Pareto Charts 1060
Example 36.3. Highlighting the “Vital Few” 1066
Example 36.4. Highlighting Combinations of Categories 1067
Example 36.5. Highlighting Combinations of Cells 1069
Example 36.6. Ordering Rows and Columns in a Comparative Pareto Chart . 1071
Example 36.7. Merging Columns in a Comparative Pareto Chart 1073
Example 36.8. Creating Weighted Pareto Charts 1075

Chapter 36
Details and Examples

Details

This chapter provides details on the following topics:

- terminology
- labels for chart features
- scaling the cumulative percent curve
- creating output data sets
- constructing effective Pareto charts
- missing values
- the role of variable formats
- large data sets

The "Examples" section illustrates these topics with several detailed examples.

Terminology

Basic Pareto Charts

A basic Pareto chart (see Figure 33.1 on page 964) analyzes the unique values of a *process variable*, which are referred to as *Pareto categories* or *levels*. These values typically represent problems encountered during some phase of a manufacturing or service activity.

A basic vertical Pareto chart (as produced by the Pareto procedure's VBAR statement) has one horizontal and two vertical axes. The horizontal (or *category*) axis is displayed at the bottom of the chart and lists the Pareto categories. The *primary vertical axis* (or *frequency axis*) is displayed on the left. The relative frequency of each Pareto category is represented by a vertical bar whose height is measured on the primary vertical axis. You can use the SCALE= option to scale this axis in percent, count, or weight units. The *secondary vertical axis* (or *cumulative percent axis*) is displayed on the right. This axis is scaled in cumulative percent units and is used to read the *cumulative percent curve*. The height of each point on the curve represents the percent of the total frequency accounted for by the Pareto categories to the left of the point.

In a horizontal Pareto chart (as produced by the HBAR statement), the category axis is displayed vertically on the left. Categories appear in order of decreasing relative frequency from top to bottom. The frequency axis appears at the top of the chart and the cumulative percent axis is at the bottom. The relative frequencies of the Pareto categories are represented by horizontal bars. A point on the cumulative percent curve

represents the percent of the total frequency accounted for by the Pareto categories above that point.

Note: For the sake of brevity, in this chapter the term *height* is used to refer to the size of a bar as measured along the frequency axis, whether the Pareto chart is oriented vertically or horizontally.

Restricted Pareto Charts

A *restricted Pareto chart* (see Figure 33.6 on page 969) displays only the n most frequently occurring categories in a data set that contains N categories, where $N > n$. The remaining $N - n$ categories are dropped or are merged into a single "other" category created with the OTHER= option. The MAXCMPCT=, MAXNCAT=, and MINPCT= options provide alternative methods for specifying n. See the entries for these options in "Dictionary of Options" on page 976.

Weighted Pareto Charts

A *weighted Pareto chart* (see Example 36.8 on page 1075) displays bars whose heights represent the weighted frequencies of the categories. Typical weights are the cost of repair or the loss incurred by the customer.

The weight W_i for the i^{th} Pareto category is computed as

$$W_i = \sum_{u \in \mathcal{C}_i} w(u) f(u)$$

where $\mathcal{C}_i$ is the set of observations that make up the i^{th} category, $w(u)$ is the value of the weight variable in the u^{th} observation, and $f(u)$ is the value of the frequency variable in the u^{th} observation (taking $f(u) \equiv 1$ if a FREQ= variable is not specified). If SCALE=WEIGHT is specified, the height of the bar for the i^{th} category is W_i. If SCALE=PERCENT is specified, the height of this bar is

$$\frac{100 W_i}{\sum_{j=1}^{N} W_j}$$

where N is the total number of categories.

Comparative Pareto Charts

A *comparative Pareto chart* combines two or more Pareto charts for the same process variable. The component charts are displayed with uniform axes to facilitate comparison. The observations represented by a components chart are referred to as a *cell*. The framed areas for the component charts are referred to as *tiles*.

In a *one-way comparative Pareto chart*, each component chart corresponds to a different level of a single classification variable specified with the CLASS= option. The component charts are arranged in a stack or a row, as illustrated in Output 36.1.3 (page 1058), Output 36.2.2 (page 1063), and Output 36.2.3 (page 1064). In a *two-way comparative Pareto chart*, each component chart corresponds to a different combination

of levels of two classification variables specified with the CLASS= option. The component charts are arranged in a matrix, as illustrated in Output 36.2.4 on page 1065.

In any comparative Pareto chart there is a *key cell*, in which the bars are in decreasing order and whose order is imposed on all the other cells to achieve a uniform category axis. By default, the key cell is the cell in the upper left corner, but you can use the CLASSKEY= option to designate any other cell as the key cell. In this case, the rows and columns of the comparative chart will be rearranged so that the key cell appears in the upper left. However, if you require the rows and columns in a particular order, you can specify the NOKEYMOVE option in conjunction with the CLASSKEY= option to suppress the rearrangement.

If you are creating your chart with a graphics device, you can use the NROWS= and NCOLS= options to specify the numbers of rows and columns in a comparative Pareto chart. By default, NROWS=2 and NCOLS=1 for a one-way comparison and NROWS=2 and NCOLS=2 for a two-way comparison. There is no upper limit to the number of rows or columns that you can specify, but in practice the limit is determined by the display area of your graphics device. If the numbers of classification variable levels exceed the NROWS= and NCOLS= values, the chart is created on multiple screens or pages.

If the same set of Pareto categories does not occur in each cell of a comparative Pareto chart, the categories are said to be *unbalanced*. In this case, the procedure uses the following convention to construct the uniform category axis. First, the categories that occur in the key cell are arranged on the category axis from left to right (top to bottom for a horizontal chart), sorted in decreasing order of frequency, with tied levels arranged in order of their formatted values. The categories not in the key cell are assigned frequencies of zero in the key cell, and they are arranged at the right (bottom) of the category axis, where they are ordered by their formatted values. This arrangement is simply a convention of the procedure and should not be interpreted to mean that one category is more important than another.

Whether the categories in the input data set are balanced or not, the categories in the OUT= data set are always balanced. The procedure balances this data set by assigning values of zero to the _COUNT_ and _PCT_ variables as necessary.

Unbalanced categories present a special problem when the MAXNCAT= option is used to restrict the number of categories displayed on the chart. For instance, suppose that you specify MAXNCAT=12 and there are 15 categories in all, 10 of which occur in the key cell. Since there is no unambiguous method for selecting two of the remaining five categories to complete the restricted list, the procedure reduces the restricted list to the categories that occur in the key cell and displays only those 10 categories. A warning message is issued in the SAS log.

Labels for Chart Features

The following table summarizes methods for labeling the features of Pareto charts.

Table 36.1. Labeling Features of Pareto Charts

Feature	Method for Specifying Label
titles	TITLE*n* statements
footnotes	FOOTNOTE*n* statements
category axis	*process variable* label
primary vertical axis (VBAR only)	VAXISLABEL= option
secondary vertical axis (VBAR only)	VAXIS2LABEL= option
primary horizontal axis (HBAR only)	HAXISLABEL= option
secondary horizontal axis (HBAR only)	HAXIS2LABEL= option
bars	BARLABEL= option
points on cumulative percent curve	CMPCTLABEL= option
rows and columns	CLASS= variable labels
cells	NLEGEND option or NLEGEND= variable
category legend	CATLEGLABEL= option
high/low bar legend	HLLEGLABEL= option
bar color legend	BARLEGLABEL= option
tile legend	TILELEGLABEL= option
annotation	ANNOTATE= and ANNOTATE2= data sets

Scaling the Cumulative Percent Curve

Pareto charts shown in textbooks typically scale the cumulative percent curve so that it is anchored at the top right corner of the leftmost bar. The upper end of the primary vertical axis is then extended to accommodate the curve. For an illustration, see Output 36.2.1 on page 1062. By default, the PARETO procedure uses the top right corner as the anchor position on a vertical chart and the bottom right corner of the topmost bar as the anchor position on a horizontal chart. You can override the default with the ANCHOR= option.

This method of scaling is not feasible if the number of categories is very large and if the Pareto distribution is uniform. In this case, the bars are excessively compressed relative to the curve. Conversely, this method excessively compresses the curve relative to the bars when you use a count scale for the frequency axis in a comparative Pareto chart and the tallest bar does not occur in the key cell. In either situation, the procedure overrides the textbook scaling method and balances the scales of the bars and the curve.

You can use the AXISFACTOR= option to specify the extent to which the frequency axis should be extended. Alternatively, you can extend the frequency axis by using

the VBAR statement VAXIS= option or HBAR statement HAXIS= option to specify the tick mark values for the axis.

Another scaling anomaly is illustrated by the comparative Pareto chart in Output 36.1.4 on page 1059. Here, the cumulative percent curve in the bottom chart is not anchored due to the combination of a uniform count scale and different sample sizes in the two cells.

Output Data Sets

The OUT= data set saves the information displayed on a Pareto chart. If you specify CLASS= variables, the OUT= data set contains one block of observations for each combination of levels of the CLASS= variables, and within each block there is an observation. The observations are sorted in the order in which the categories are displayed on the chart. The following variables read from a DATA= data set are saved in an OUT= data set:

- process variables
- CLASS= variables
- BY variables
- WEIGHT= variables
- the CTILES= variable
- the TILELEGEND= variable
- the NLEGEND= variable
- CBARS= variables
- PBARS= variables
- BARLEGEND= variables

In addition, the OUT= data set contains the following variables that are created during the analysis:

- _COUNT_, which saves the frequency count for each Pareto category
- _WCOUNT_, which saves the weighted count for each category. This variable is created only when you specify the WEIGHT= option.
- _PCT_, which saves the percent of the total count for each category. If you specify the WEIGHT= option, the variable _PCT_ saves the percent of the total weighted count.
- _CMPCT_, which saves the cumulative percent for each category

See Output 36.8.2 on page 1076 for an example of an OUT= data set.

If you specify the MAXNCAT=, MAXCMPCT=, or MINPCT= option, the OUT= data set saves only the categories displayed on the chart. If you create an OTHER= category that merges the remaining categories, an additional observation is saved with the new category. Since the OTHER= value is defined as a formatted value of the process variable, you should also specify a corresponding internal value, as follows:

- If the process variable is a character variable, specify the internal value with the OTHERCVAL= option. If you do not specify this value, the OTHER= value is saved as the internal value.
- If the process variable is a numeric variable, specify the internal value with the OTHERNVAL= option. If you do not specify this value, an internal missing value is saved.

Constructing Effective Pareto Charts

The following are recommendations for improving the visual clarity of Pareto charts:

- Decide carefully how the bars should be scaled. The default percent scale is not always the best choice. For instance, a count scale may be more appropriate in a comparative Pareto chart where the total count per cell varies widely from cell to cell and where you want to compare Pareto distributions on an *absolute* scale rather than a *relative* scale. You can request a count scale by specifying SCALE=COUNT. In other situations, it may be more appropriate to use a weighted percent scale or a weighted count scale (specify a WEIGHT= variable and either SCALE=PERCENT or SCALE=WEIGHT).
- Use a weight variable if the counts are dependent on a factor such as exposure or opportunity that varies from one category to another. For instance, suppose that you are creating a Pareto chart for the number of medical claims submitted by company employees categorized by job title. The counts can be weighted to adjust for the fact that there are more individuals in some jobs than in others and for the fact that some jobs may be associated with greater health risks than others.
- Use the NOCURVE option to eliminate the cumulative percent curve in situations where the curve reveals little information about the data. In general, the bars should be more prominent than the curve.
- Maximize the space used for the bars by eliminating unnecessary labels and visual clutter. This is particularly important for comparative Pareto charts. The NOHLABEL and NOVLABEL options are useful for this purpose. You can also use the NOVLABEL2, NOVTICK, and NOVTICK2 options with a VBAR statement or the NOHLABEL2, NOHTICK and NOHTICK2 options with an HBAR statement.
- Make legends more informative by specifying legend labels.
- Avoid filling bars with multiple types of cross-hatched patterns; solid color fills are less distracting. Use color sparingly to emphasize important features (such as the “vital few” categories), and choose bar colors that provide good visual discrimination.
- If you are working with a large data set involving many categories, limit the number displayed to achieve visual clarity.
- If your application involves classification effects, construct more than one Pareto chart for the data using various combinations of classification variables (this approach is illustrated in Example 36.2 on page 1060.
- Provide reference lines on comparative Pareto charts to aid visual comparison.

Refer to Chapter 2 of Cleveland (1985) for a general discussion of the principles of statistical graphics.

Missing Values

By default, observations with missing values of a process variable are not processed. If you specify the MISSING option, then missing values are treated as a Pareto category.

Likewise, observations with missing values of the CLASS= variables are not processed by default. Missing values of the first CLASS= variable are treated as a level if the MISSING1 option is specified, and missing values of the second CLASS= variable are treated as a level if the MISSING2 option is specified.

Role of Variable Formats

The categories of a Pareto chart are always determined using formatted values of the process variable, and the format is used to label the categories.

On the chart, the categories are displayed in decreasing order of frequency. If there are multiple categories with the same count, the tied categories are displayed in order of their formatted values.

When you create a comparative Pareto chart, the formatted levels of the CLASS= variables are used to group the data into cells. There is a cell for each level of the CLASS= variable in a one-way comparative chart, and there is a cell for each combination of levels of the CLASS= variables in a two-way comparative chart.

You can specify the order of the rows and columns corresponding to the classification levels with the ORDER1= and ORDER2= options. The default value of these options is INTERNAL, which means that the order is determined by the internal values of the CLASS= variables. It is possible for a particular formatted value to correspond to more than one internal value. To resolve this ambiguity, the internal value that determines the position of the row or column is the value that occurs first in the input data set.

Other values that you can specify for the ORDER1= and ORDER2= options are FORMATTED, FREQ, and DATA. Detailed descriptions of these options are provided on pages 990–991.

Large Data Sets

While there is no limit to the number of observations that can be read from an input data set, the maximum number of Pareto categories that can be read is 32,767. This limit is a practical issue only if you are creating a restricted Pareto chart from a large data set, since the number of categories that can be displayed is limited by the resolution of your graphics device. The number of categories that can be read is limited by the amount of memory available, since the levels are stored in memory. If you run out of memory, you should first reduce the data with the FREQ procedure.

Examples

Example 36.1. Creating Before-and-After Pareto Charts

See PARETO7 in the SAS/QC Sample Library

During the manufacture of a metal-oxide semiconductor (MOS) capacitor, causes of failures were recorded before and after a tube in the diffusion furnace was cleaned. This information was saved in a SAS data set named FAILURE3.

```
data failure3;
   length cause $ 16 stage $ 16 ;
   label  cause = 'Cause of Failure' ;
   input  stage $ 3-18 cause $ 21-36 counts;
   datalines;
  Before Cleaning   Contamination    14
  Before Cleaning   Corrosion         2
  Before Cleaning   Doping            1
  Before Cleaning   Metallization     2
  Before Cleaning   Miscellaneous     3
  Before Cleaning   Oxide Defect      8
  Before Cleaning   Silicon Defect    1
  After Cleaning    Doping            0
  After Cleaning    Corrosion         2
  After Cleaning    Metallization     4
  After Cleaning    Miscellaneous     2
  After Cleaning    Oxide Defect      1
  After Cleaning    Contamination    12
  After Cleaning    Silicon Defect    2
run;
```

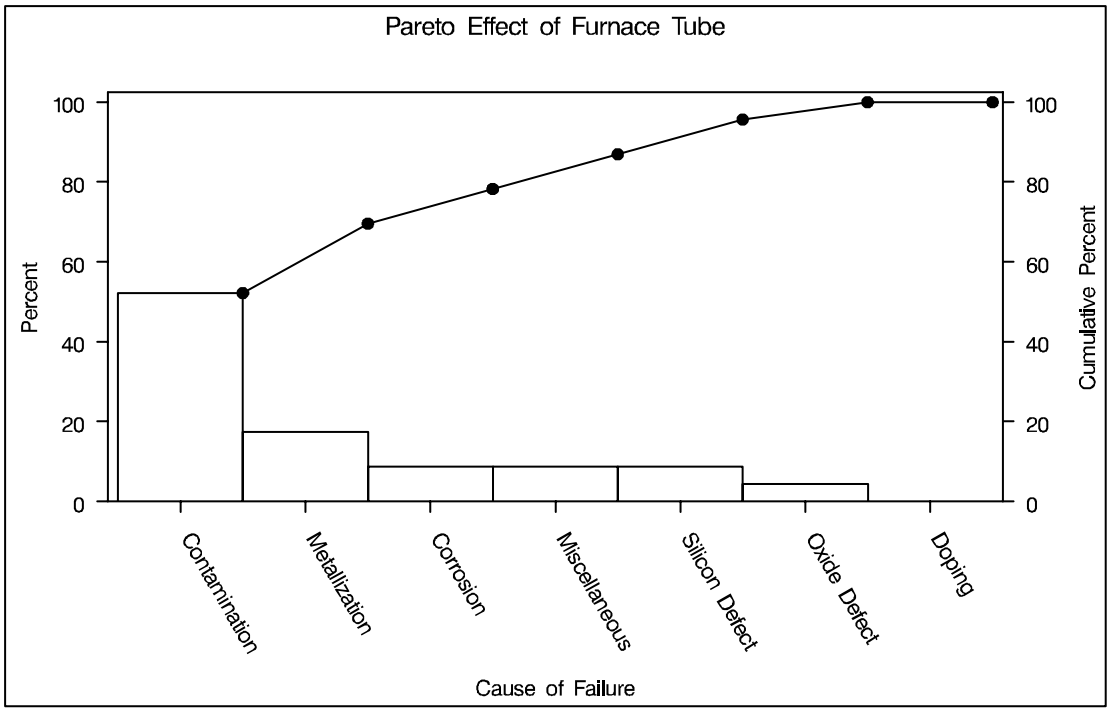
To compare distribution of failures before and after cleaning, you can create two separate Pareto charts, one for the observations in which STAGE is equal to `Before Cleaning` and one for the observations in which STAGE is equal to `After Cleaning`. You can do this with the BY statement.

```
proc sort data=failure3;
   by stage;
run;

title 'Pareto Effect of Furnace Tube' ;
proc pareto data=failure3;
   vbar cause / freq       = counts
                angle      = -60;
   by stage;
run;
```

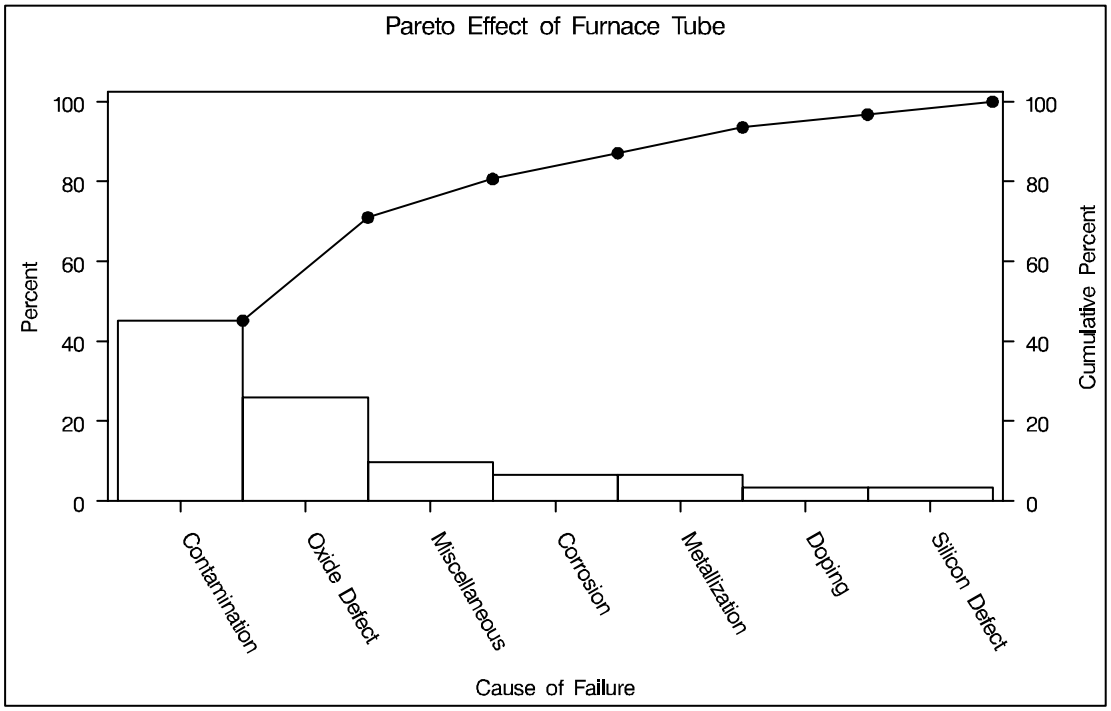
The SORT procedure sorts the observations in order of the values of STAGE. It is not necessary to sort by the values of CAUSE since this is done by the PARETO procedure. The two charts, displayed in Output 36.1.1 and Output 36.1.2, reveal a

reduction in oxide defects after the tube was cleaned. This is a relative reduction, since the primary axes are scaled in percent units.

Output 36.1.1. "After" Analysis Using STAGE as a BY Variable



Output 36.1.2. "Before" Analysis Using STAGE as a BY Variable



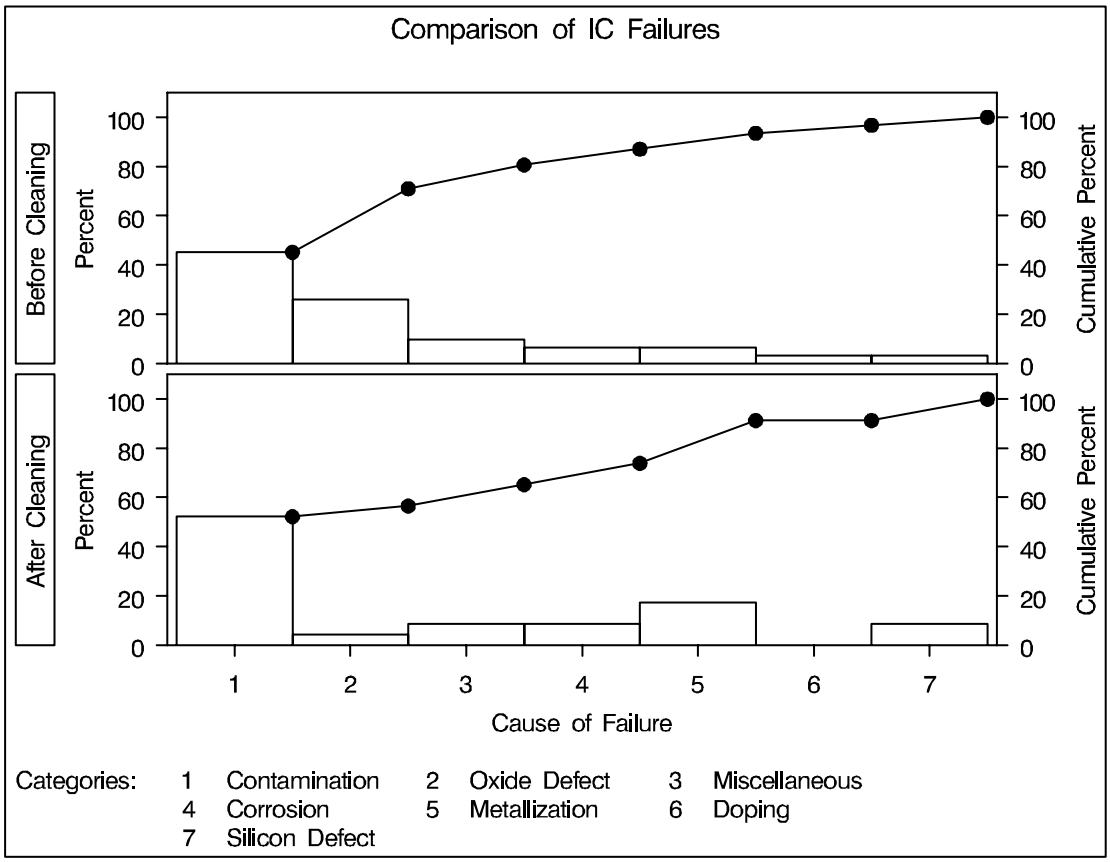
In general, it is difficult to compare Pareto charts created with BY processing because their axes are not necessarily uniform. A better approach is to construct a comparative Pareto chart, as illustrated by the following statements:

```
title 'Comparison of IC Failures' ;
proc pareto data=failure3;
   vbar cause / class      = stage
                freq       = counts
                scale      = percent
                intertile  = 1.0
                classkey   = 'Before Cleaning' ;
run;
```

See PARETO8 in the SAS/QC Sample Library

The CLASS= option designates STAGE as a classification variable, and this directs the procedure to create the one-way comparative Pareto chart, shown in Output 36.1.3, that displays a component chart for each level of STAGE.

Output 36.1.3. Before-and-After Analysis Using Comparative Pareto Chart



In a comparative Pareto chart, there is always one special cell, called the *key cell*, in which the bars are displayed in decreasing order, and whose order determines the uniform horizontal axis used for all the cells. The key cell is positioned at the top of the chart. Here, the key cell is the set of observations for which STAGE equals

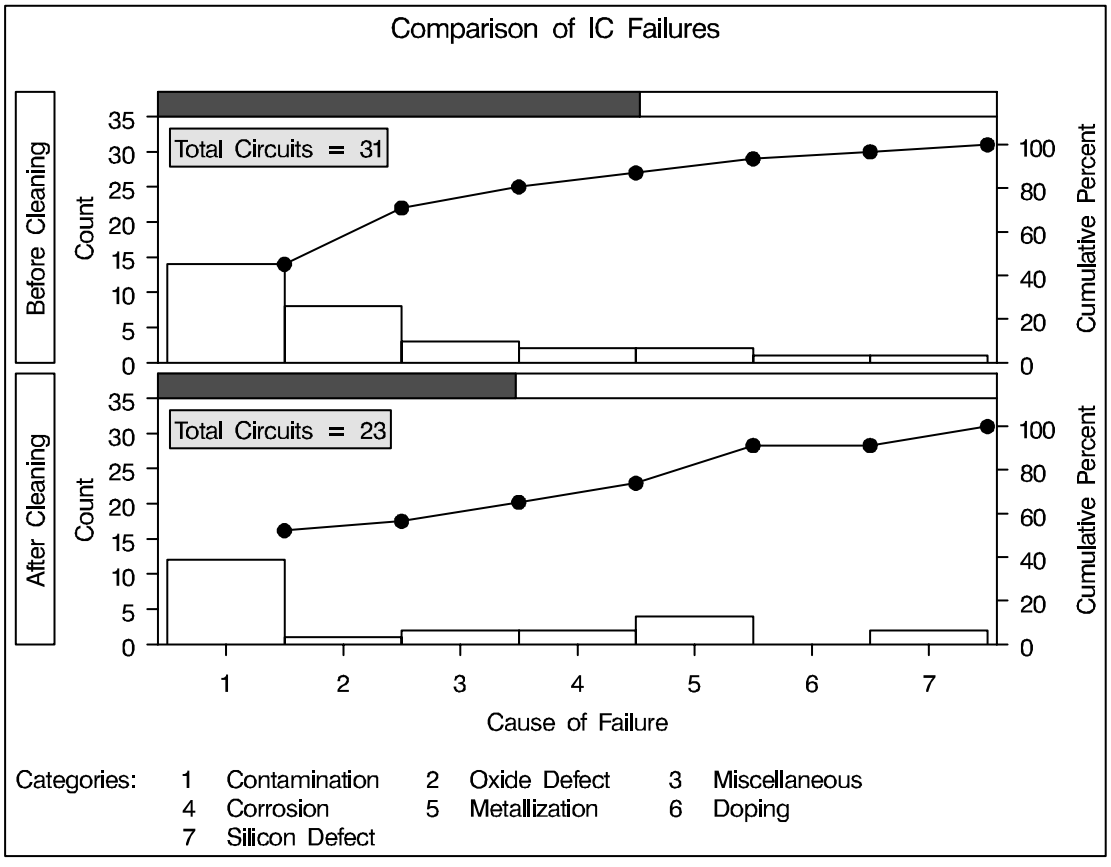
`Before Cleaning`, as specified by the CLASSKEY= option. By default, the levels are sorted in the order determined by the ORDER1= option, and the key cell is the the level that occurs first in this order.

In many applications, it may be more revealing to base comparisons on counts rather than percents. The following statements construct a chart with a frequency scale:

```
title 'Comparison of IC Failures' ;
proc pareto data=failure3;
   vbar cause / class      = stage
                freq       = counts
                scale      = count
                intertile  = 1.0
                nlegend    = 'Total Circuits'
                cframenleg = yellow
                cprop      = red
                classkey   = 'Before Cleaning' ;
run;
```

The chart is shown in Output 36.1.4.

Output 36.1.4. Before-and-After Analysis Using Comparative Pareto Chart



Specifying SCALE=COUNT scales the primary vertical axis in frequency units. The NLEGEND= option adds a sample size legend, and the CFRAMENLEG= option

frames the legend. The CPROP= option adds bars that indicate the proportion of total frequency represented by each cell. The INTERTILE= option separates the tiles with a small offset.

Note that the lower cumulative percent curve in Output 36.1.4 is not anchored to the first bar. This is a consequence of the uniform frequency scale and of the fact that the number of observations in each cell is not the same.

Example 36.2. Creating Two-Way Comparative Pareto Charts

See PARETO9 in the SAS/QC Sample Library

During the manufacture of a MOS capacitor, different cleaning processes were used by two manufacturing systems operating in parallel. Process A used a standard cleaning solution, while Process B used a different cleaning mixture that contained less particulate matter. The failure causes observed with each process for five consecutive days were recorded and saved in a SAS data set called FAILURE4.

```
data failure4;
   label  cause   = 'Cause of Failure' ;
   input  process $ 1-9 day $ 13-19 cause $ 23-36 counts 40-41;
   datalines;
Process A   March 1   Contamination    15
Process A   March 1   Corrosion         2
Process A   March 1   Doping            1
Process A   March 1   Metallization     2
Process A   March 1   Miscellaneous     3
Process A   March 1   Oxide Defect      8
Process A   March 1   Silicon Defect    1
Process A   March 2   Contamination    16
Process A   March 2   Corrosion         3
Process A   March 2   Doping            1
Process A   March 2   Metallization     3
Process A   March 2   Miscellaneous     1
Process A   March 2   Oxide Defect      9
Process A   March 2   Silicon Defect    2
Process A   March 3   Contamination    20
Process A   March 3   Corrosion         1
Process A   March 3   Doping            1
Process A   March 3   Metallization     0
Process A   March 3   Miscellaneous     3
Process A   March 3   Oxide Defect      7
Process A   March 3   Silicon Defect    2
Process A   March 4   Contamination    12
Process A   March 4   Corrosion         1
Process A   March 4   Doping            1
Process A   March 4   Metallization     0
Process A   March 4   Miscellaneous     0
Process A   March 4   Oxide Defect     10
Process A   March 4   Silicon Defect    1
Process A   March 5   Contamination    23
Process A   March 5   Corrosion         1
Process A   March 5   Doping            1
Process A   March 5   Metallization     0
Process A   March 5   Miscellaneous     1
Process A   March 5   Oxide Defect      8
Process A   March 5   Silicon Defect    2
Process B   March 1   Contamination     8
Process B   March 1   Corrosion         2
```

```
Process B   March 1   Doping            1
Process B   March 1   Metallization     4
Process B   March 1   Miscellaneous     2
Process B   March 1   Oxide Defect     10
Process B   March 1   Silicon Defect    3
Process B   March 2   Contamination     9
Process B   March 2   Corrosion         0
Process B   March 2   Doping            1
Process B   March 2   Metallization     2
Process B   March 2   Miscellaneous     4
Process B   March 2   Oxide Defect      9
Process B   March 2   Silicon Defect    2
Process B   March 3   Contamination     4
Process B   March 3   Corrosion         1
Process B   March 3   Doping            1
Process B   March 3   Metallization     0
Process B   March 3   Miscellaneous     0
Process B   March 3   Oxide Defect     10
Process B   March 3   Silicon Defect    1
Process B   March 4   Contamination     2
Process B   March 4   Corrosion         2
Process B   March 4   Doping            1
Process B   March 4   Metallization     0
Process B   March 4   Miscellaneous     3
Process B   March 4   Oxide Defect      7
Process B   March 4   Silicon Defect    1
Process B   March 5   Contamination     1
Process B   March 5   Corrosion         3
Process B   March 5   Doping            1
Process B   March 5   Metallization     0
Process B   March 5   Miscellaneous     1
Process B   March 5   Oxide Defect      8
Process B   March 5   Silicon Defect    2
;
run;
```

In addition to the process variable CAUSE, there are two classification variables in this data set: PROCESS and DAY. The variable COUNTS is a frequency variable.

This example creates a series of displays that progressively use more of the classification information.

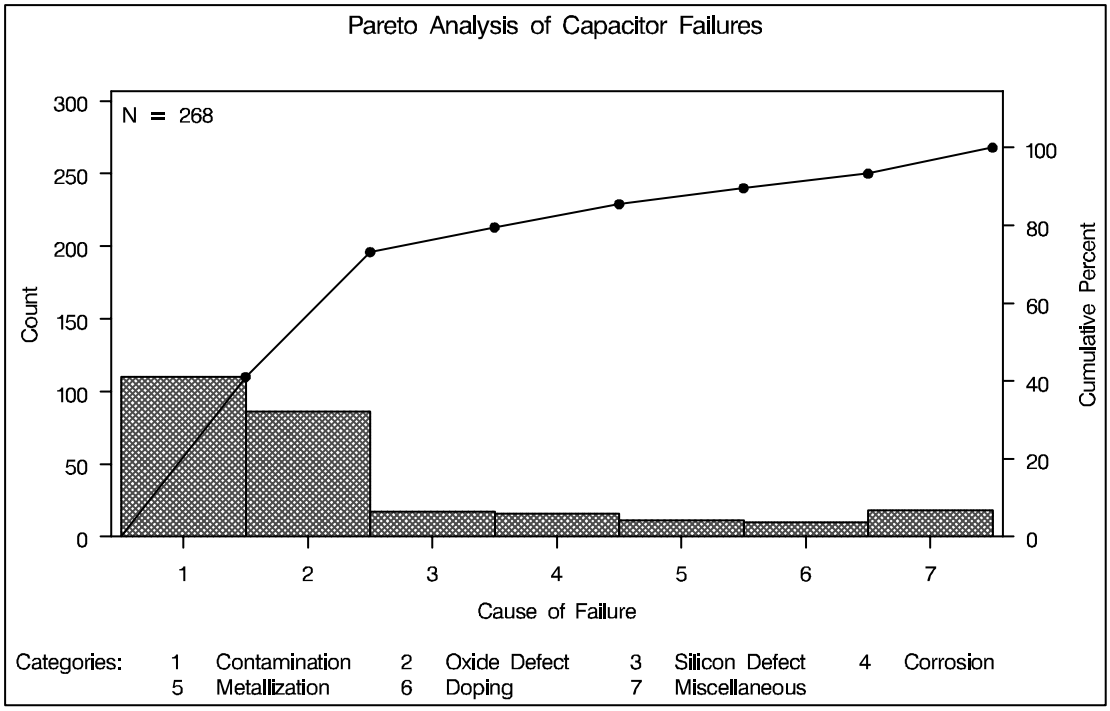
Basic Pareto Chart

The first display, created with the following statements, analyzes the process variable without taking into account the classification variables.

```
title 'Pareto Analysis of Capacitor Failures' ;
proc pareto data=failure4;
   vbar cause / freq        = counts
                last        = 'Miscellaneous'
                scale       = count
                anchor      = bl
                cbars       = green
                pbars       = m5x45
                nlegend ;
run;
```

The chart, shown in Output 36.2.1, indicates that contamination is the most frequently occurring problem.

Output 36.2.1. Pareto Analysis without Classification Variables



The color and pattern for the bars are specified with the CBARS= and PBARS= options. The pattern M5X45 is a particular type of crosshatching (refer to *SAS/GRAPH Software: Reference* for a pattern selection guide). If you specify a color but not a pattern, the bars are filled with a solid color.

The option ANCHOR=BL anchors the cumulative percent curve at the bottom left (BL) of the first bar. The NLEGEND option adds a sample size legend.

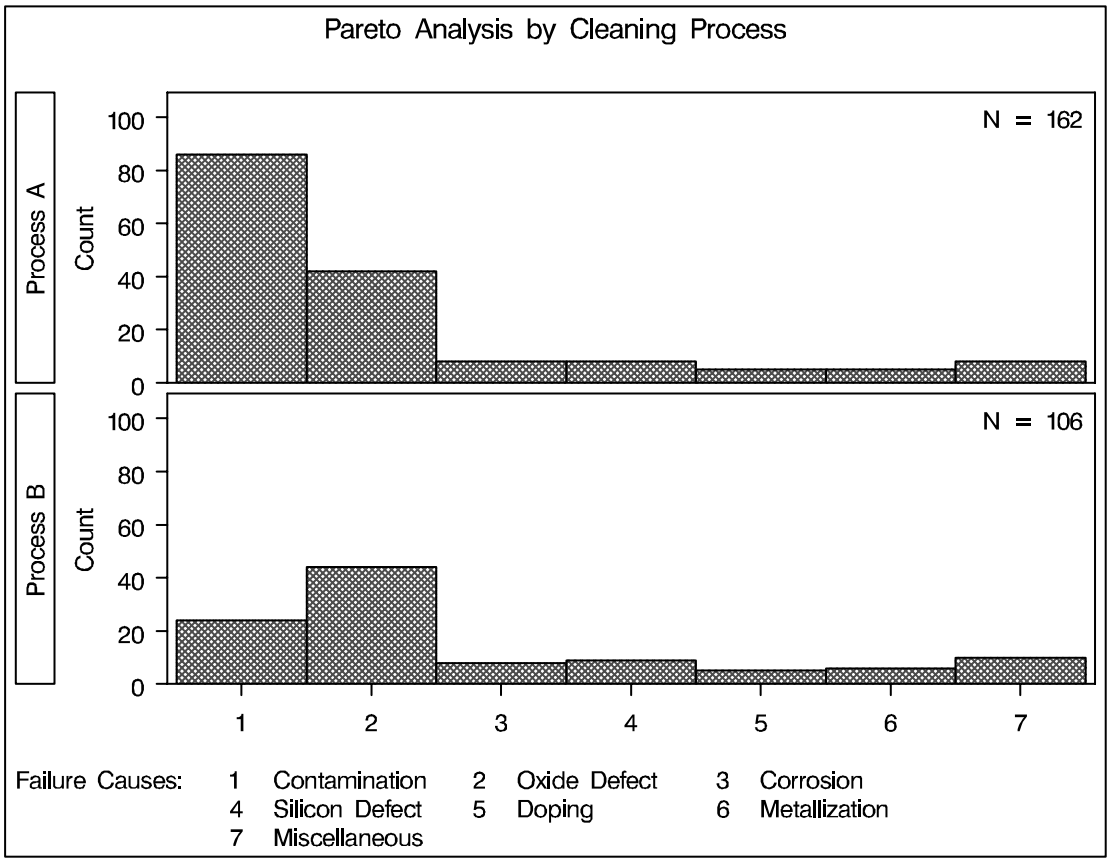
One-Way Comparative Pareto Chart for PROCESS

The following statements specify PROCESS as a classification variable to create the comparative Pareto chart displayed in Output 36.2.2:

```
title 'Pareto Analysis by Cleaning Process' ;
proc pareto data=failure4;
   vbar cause / class       = process
                freq        = counts
                last        = 'Miscellaneous'
                scale       = count
                catleglabel = 'Failure Causes:'
                intertile   = 1.0
                cbars       = green
```

```
                    pbars       = m5x45
                    nohlabel
                    nocurve
                    nlegend ;
run;
```

Output 36.2.2. One-Way Comparative Pareto Analysis with CLASS=PROCESS



Each cell corresponds to a level of the CLASS= variable (PROCESS). By default, the cells are arranged from top to bottom in alphabetical order of the formatted values of PROCESS, and the key cell is the top cell. The main difference in the two cells is a drop in contamination using Process B.

The CATLEGLABEL= option specifies the category legend label *Failure Causes:*. The NOHLABEL option suppresses the horizontal axis labels. The NOCURVE option suppresses the cumulative percent curve.

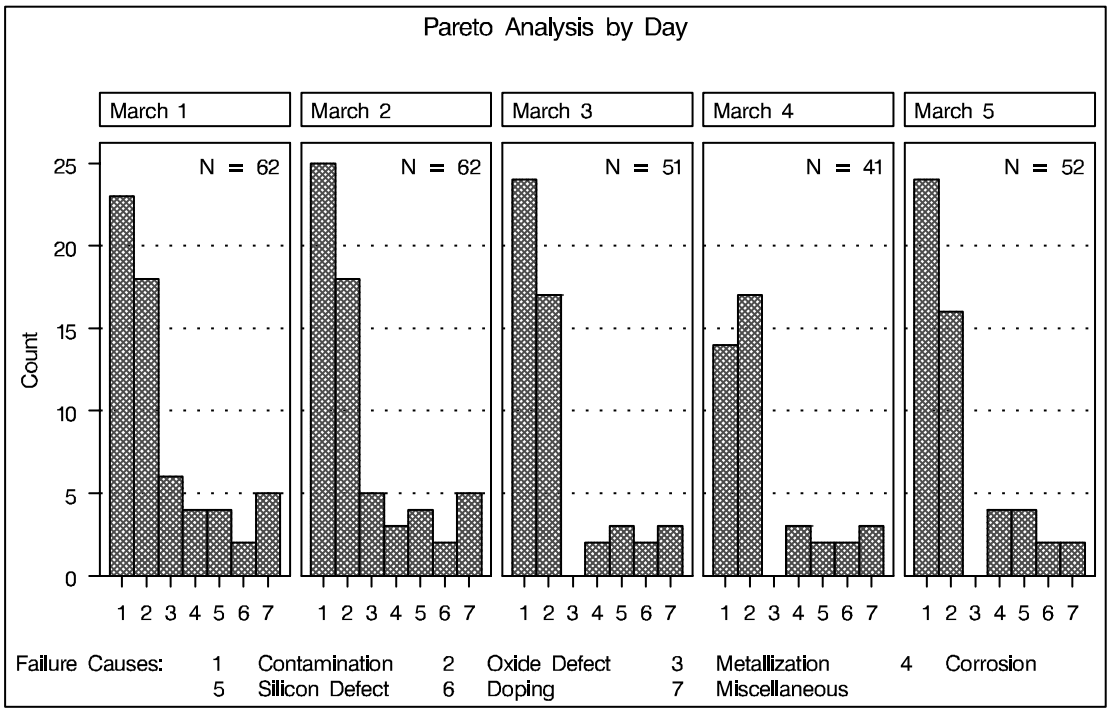
One-way Comparative Pareto Chart for DAY

The following statements specify DAY as a classification variable:

```
title 'Pareto Analysis by Day';
proc pareto data=failure4;
   vbar cause / class       = day
                freq        = counts
                last        = 'Miscellaneous'
                scale       = count
                cbars       = green
                pbars       = m5x45
                catleglabel = 'Failure Causes:'
                intertile   = 1.0
                nrows       = 1
                ncols       = 5
                vref        = 5 10 15 20
                lvref       = 34
                nohlabel
                nocurve
                nlegend ;
run;
```

The NROWS= and NCOLS= options display the cells in a side-by-side arrangement. The VREF= and LVREF= options add reference lines. The chart is displayed in Output 36.2.3.

Output 36.2.3. One-Way Comparative Pareto Analysis with CLASS=DAY



By default, the key cell is the leftmost cell. There were no failures due to Metallization starting on March 3 (in fact, process controls to reduce this problem were introduced on this day).

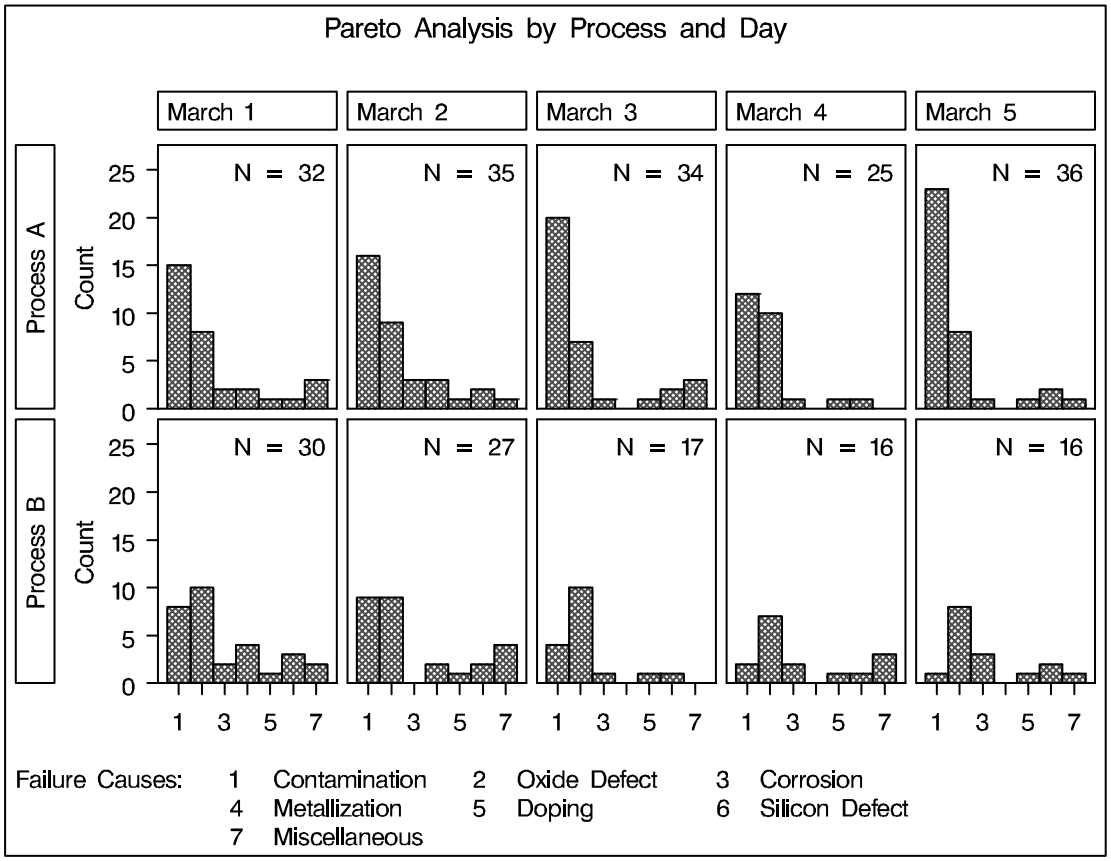
Two-way Comparative Pareto Chart for PROCESS and DAY

The following statements specify both PROCESS and DAY as CLASS= variables to create a two-way comparative Pareto chart:

```
title 'Pareto Analysis by Process and Day' ;
proc pareto data=failure4;
   vbar cause / class       = ( process day )
                freq        = counts
                nrows       = 2
                ncols       = 5
                cbars       = green
                pbars       = m5x45
                last        = 'Miscellaneous'
                scale       = count
                catleglabel = 'Failure Causes:'
                intertile   = 1.0
                nohlabel
                nocurve
                nlegend ;
run;
```

The chart is displayed in Output 36.2.4.

Output 36.2.4. Two-Way Comparative Pareto Analysis for PROCESS and DAY



The cells are arranged in a matrix whose rows correspond to levels of the first CLASS= variable (PROCESS) and whose columns correspond to levels of the second CLASS= variable (DAY). The dimensions of the matrix are specified with the NROWS= and NCOLS= options. The key cell is in the upper left corner.

The chart reveals continuous improvement with Process B.

Example 36.3. Highlighting the "Vital Few"

See PARETO10 in the SAS/QC Sample Library

This example is a continuation of Example 36.2.

In some applications you may want to use colors and patterns to highlight the bars corresponding to the most frequently occurring categories, which are referred to as the "vital few."

The following statements highlight the two most frequently occurring categories in each cell of the comparative Pareto chart shown in Output 36.2.4:

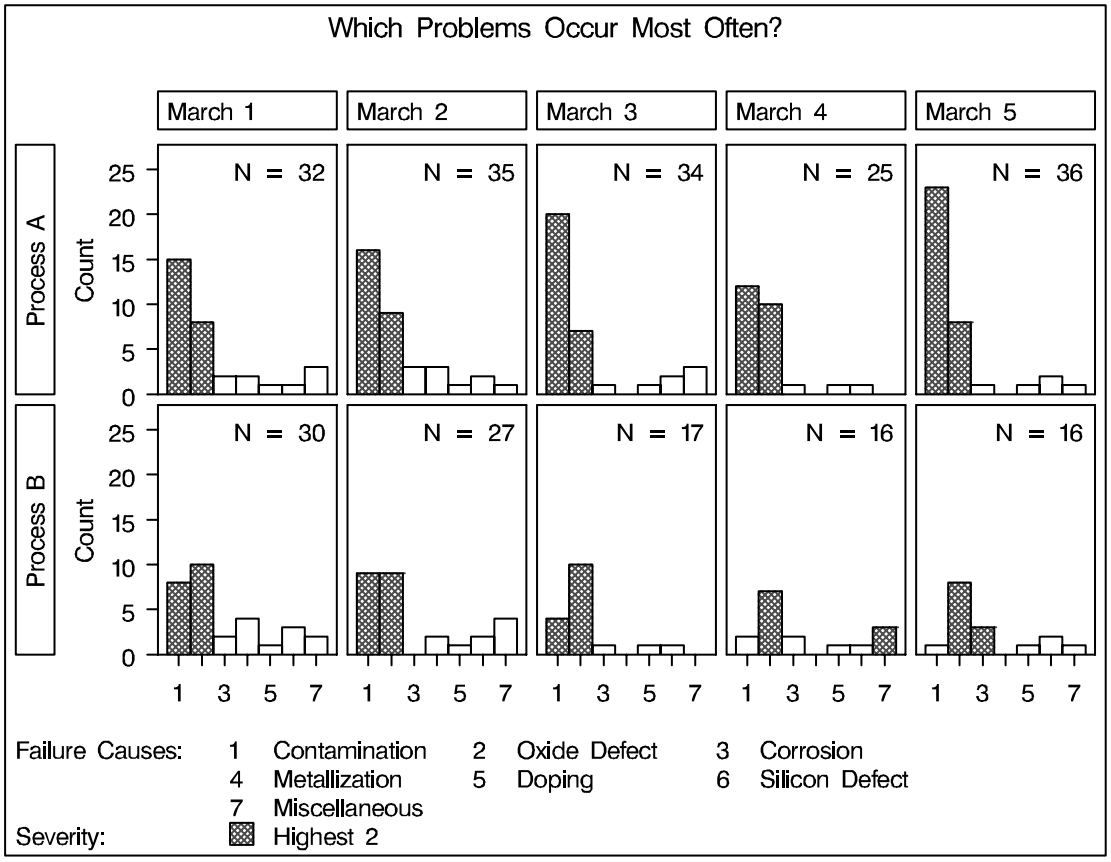
```
title 'Which Problems Occur Most Often?';
proc pareto data=failure4;
   vbar cause / class       = ( process day )
                freq        = counts
                nrows       = 2
                ncols       = 5
                last        = 'Miscellaneous'
                scale       = count
                chigh(2)    = green
                phigh(2)    = m5x45
                hlleglabel  = 'Severity:'
                catleglabel = 'Failure Causes:'
                intertile   = 1.0
                nohlabel
                nocurve
                nlegend ;
run;
```

Specifying CHIGH(2)=GREEN and PHIGH(2)=M5X45 causes the two highest bars in each cell to be filled in green with the pattern M5X45 (refer to *SAS/GRAPH Software: Reference* for a pattern selection guide). If you omit the PHIGH(2)= option, a solid green fill is used.

The new chart is displayed in Output 36.3.1. In all but two of the cells, the two vital problems are Contamination and Oxide Defect.

You can also highlight the "trivial many" categories (also referred to as the "useful many") with the CLOW(m)= and PLOW(m)= options. You can use these options in conjunction with the CHIGH(n)=, PHIGH(n)=, CBARS=, and PBARS= options. For further details, see the entries for these options in the "Dictionary of Options" on page 976.

Output 36.3.1. Emphasizing the “Vital Few”



Example 36.4. Highlighting Combinations of Categories

See PARETO11 in the SAS/QC Sample Library

In some applications, it is useful to classify the categories into groups that are not necessarily related to frequency. This example, which is a continuation of Example 36.2, shows how you can display this classification with a bar legend.

Suppose that Contamination and Metallization are high priority problems, Oxide Defect is a medium priority problem, and all other categories are low priority problems. Begin by adding this information to the data set FAILURE4.

```
data failure4;
   length color $ 8 pattern $ 8 priority $ 16 ;
   set failure4;
if cause='Contamination' or cause='Metallization' then do;
   color='red';    pattern='s';     priority='High';   end;
else if cause='Oxide Defect' then do;
   color='yellow'; pattern='m5x45'; priority='Medium'; end;
else do;
   color='white';  pattern='s';     priority='Low' ;   end;
run;
```

The variable PRIORITY indicates the priority, and the variables COLOR and PATTERN (character variables of length eight) provide colors and patterns corresponding to the levels of PRIORITY. The pattern values `S` and `M5X45` correspond to a solid fill and a crosshatched fill, respectively.

The following statements specify PRIORITY as a BARLEGEND= variable, COLOR as a CBARS= variable, and PATTERN as a PBARS= variable:

```
title 'Which Problems Take Priority?';
proc pareto data=failure4;
   vbar cause / class       = ( process day )
                freq        = counts
                nrows       = 2
                ncols       = 5
                last        = 'Miscellaneous'
                scale       = count
                cbars       = ( color )
                pbars       = ( pattern )
                barlegend   = ( priority )
                barleglabel = 'Priority:'
                catleglabel = 'Failure Causes:'
                intertile   = 1.0
                nohlabel
                nocurve
                nlegend ;
run;
```

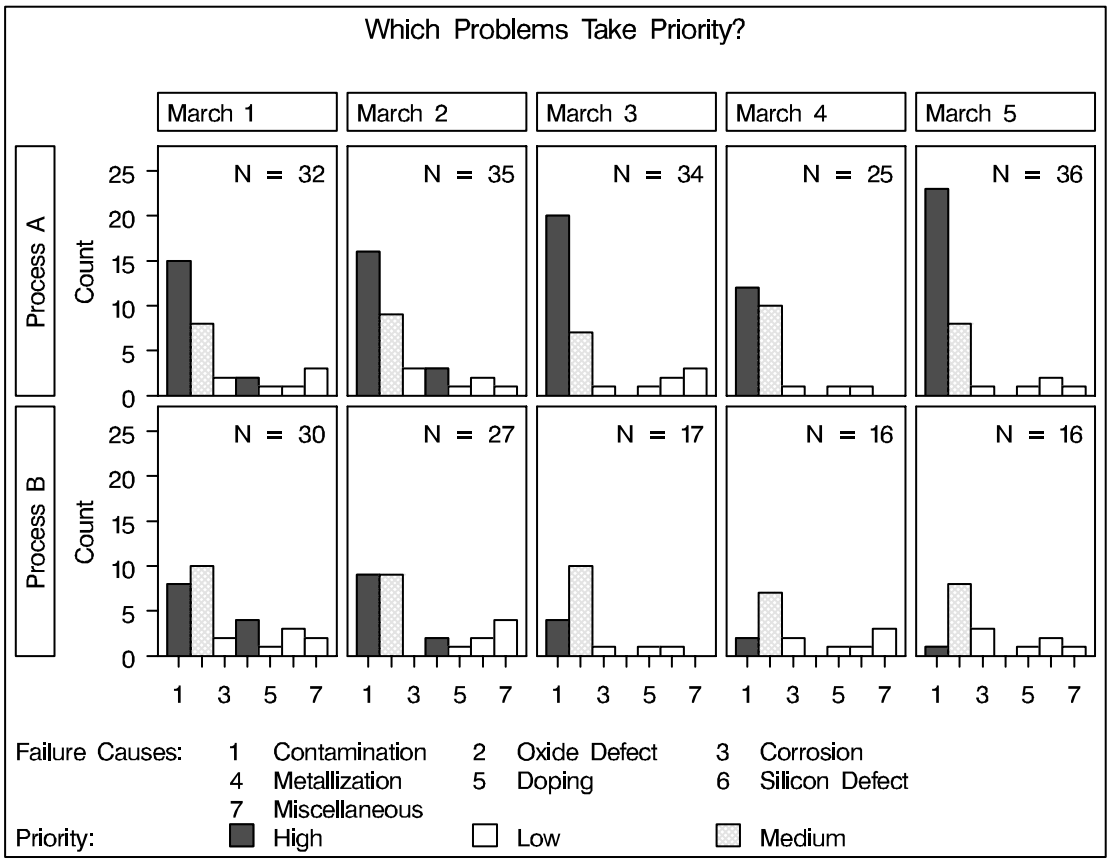
Note that the BARLEGEND=, CBARS=, and PBARS= variable names are enclosed in parentheses. (Parentheses are not used when you specify fixed colors and patterns with the CBARS= and PBARS= options, as in Example 36.2.)

The chart is displayed in Output 36.4.1. The levels of the BARLEGEND= variable are the values displayed in the legend labeled *Priority:* at the bottom of the chart.

In general, when you create CBARS=, PBARS=, and BARLEGEND= variables, their values must be consistent and unambiguous. You must assign distinct color and pattern values to the CBARS= and PBARS= variables for each level of the BARLEGEND= variable. It is not necessary to specify a PBARS= variable to accompany a BARLEGEND= variable, and if a PBARS= variable is omitted, the bars are filled with solid colors.

For further details, see the entries for the BARLEGEND=, CBARS=, and PBARS= options in "Dictionary of Options" on page 976.

Output 36.4.1. Highlighting Selected Subsets of Categories



Example 36.5. Highlighting Combinations of Cells

This example is a continuation of Example 36.4.

In some applications involving comparative Pareto charts, it is useful to classify the cells into groups. This example shows how you can display this type of classification by coloring the tiles and adding a legend.

See PARETO1 in the SAS/QC Sample Library

Suppose that you want to enhance Output 36.4.1 by highlighting the two cells for which PROCESS=`Process B` and DAY=`March 4` and `March 5` to emphasize the improvement displayed in those cells. Begin by adding a tile color variable (TILECOL) and a tile legend variable (TILELEG) to the data set FAILURE4.

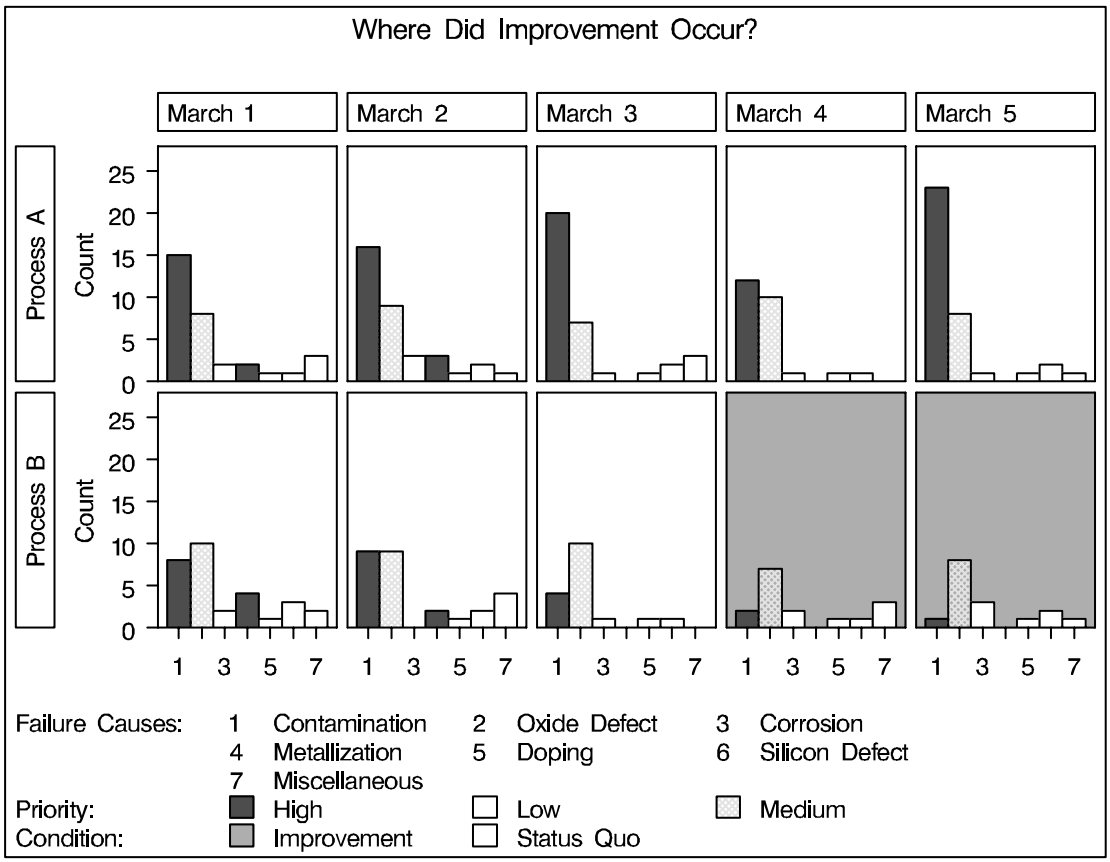
```
data failure4;
   length tilecol $ 8 tileleg $ 16 ;
   set failure4;
   if (process='Process B') and (day='March 4' or day='March 5')
   then do; tilecol='orange'; tileleg = 'Improvement'; end;
   else do; tilecol='empty' ; tileleg = 'Status Quo' ; end;
run;
```

The following statements specify TILECOL as a CTILES= variable and TILELEG as a TILELEGEND= variable. Note that the variable names are enclosed in parentheses.

```
title 'Where Did Improvement Occur?';
proc pareto data=failure4;
   vbar cause / class         = ( process day )
                freq          = counts
                nrows         = 2
                ncols         = 5
                last          = 'Miscellaneous'
                scale         = count
                catleglabel   = 'Failure Causes:'
     /* options for highlighting bars: */
                cbars         = ( color )
                pbars         = ( pattern )
                barlegend     = ( priority )
                barleglabel   = 'Priority:'
     /* options for highlighting tiles: */
                ctiles        = ( tilecol )
                tilelegend    = ( tileleg )
                tileleglabel  = 'Condition:'
                intertile     = 1.0
                nohlabel
                nocurve ;
run;
```

In the chart, shown in Output 36.5.1, the values displayed in the legend labeled *Condition:* are the levels of the TILELEGEND= variable.

Output 36.5.1. Highlighting Specific Tiles



Example 36.6. Ordering Rows and Columns in a Comparative Pareto Chart

This example illustrates methods for controlling the order of rows and columns in a comparative Pareto chart.

See PARETO13 in the SAS/QC Sample Library

The following statements create a data set named FAILURE7:

```
proc format;
   value procfmt   1   = 'Process A'
                   2   = 'Process B' ;
   value dayfmt    1   = 'Monday'
                   2   = 'Tuesday'
                   3   = 'Wednesday'
                   4   = 'Thursday'
                   5   = 'Friday'    ;
run;

data failure7;
   length cause $16 ;
   format process procfmt. day dayfmt. ;
   label  cause   = 'Cause of Failure'
          process = 'Cleaning Method'
          day     = 'Day of Manufacture' ;
   input  process day cause $16. counts @@;
   datalines;
   1   1   Contamination    15   1   1   Corrosion         2
   1   1   Doping            1   1   1   Metallization     2
   1   1   Miscellaneous     3   1   1   Oxide Defect      8
   1   1   Silicon Defect    1   1   2   Contamination    16
   1   2   Corrosion         3   1   2   Doping            1
   1   2   Metallization     3   1   2   Miscellaneous     1
   1   2   Oxide Defect      9   1   2   Silicon Defect    2
   1   3   Contamination    20   1   3   Corrosion         1
   1   3   Doping            1   1   3   Metallization     0
   1   3   Miscellaneous     3   1   3   Oxide Defect      7
   1   3   Silicon Defect    2   1   4   Contamination    12
   1   4   Corrosion         1   1   4   Doping            1
   1   4   Metallization     0   1   4   Miscellaneous     0
   1   4   Oxide Defect     10   1   4   Silicon Defect    1
   1   5   Contamination    23   1   5   Corrosion         1
   1   5   Doping            1   1   5   Metallization     0
   1   5   Miscellaneous     1   1   5   Oxide Defect      8
   1   5   Silicon Defect    2   2   1   Contamination     8
   2   1   Corrosion         2   2   1   Doping            1
   2   1   Metallization     4   2   1   Miscellaneous     2
   2   1   Oxide Defect     10   2   1   Silicon Defect    3
   2   2   Contamination     9   2   2   Corrosion         0
   2   2   Doping            1   2   2   Metallization     2
   2   2   Miscellaneous     4   2   2   Oxide Defect      9
   2   2   Silicon Defect    2   2   3   Contamination     4
   2   3   Corrosion         1   2   3   Doping            1
   2   3   Metallization     0   2   3   Miscellaneous     0
   2   3   Oxide Defect     10   2   3   Silicon Defect    1
   2   4   Contamination     2   2   4   Corrosion         2
   2   4   Doping            1   2   4   Metallization     0
   2   4   Miscellaneous     3   2   4   Oxide Defect      7
   2   4   Silicon Defect    1   2   5   Contamination     1
```

```
   2   5   Corrosion          3   2   5   Doping             1
   2   5   Metallization      0   2   5   Miscellaneous      1
   2   5   Oxide Defect       8   2   5   Silicon Defect     2
;
run;
```

Note that FAILURE7 is similar to the data set FAILURE4 created in Example 36.2. Here, the classification variables PROCESS and DAY are numeric formatted variables, and the formatted values of DAY are `Monday` through `Friday`. In Example 36.2, PROCESS and DAY are character variables, and the values of DAY are `March 1` through `March 5`.

The following statements create a two-way comparative Pareto chart for CAUSE in which the rows represent levels of PROCESS and the columns represent levels of DAY:

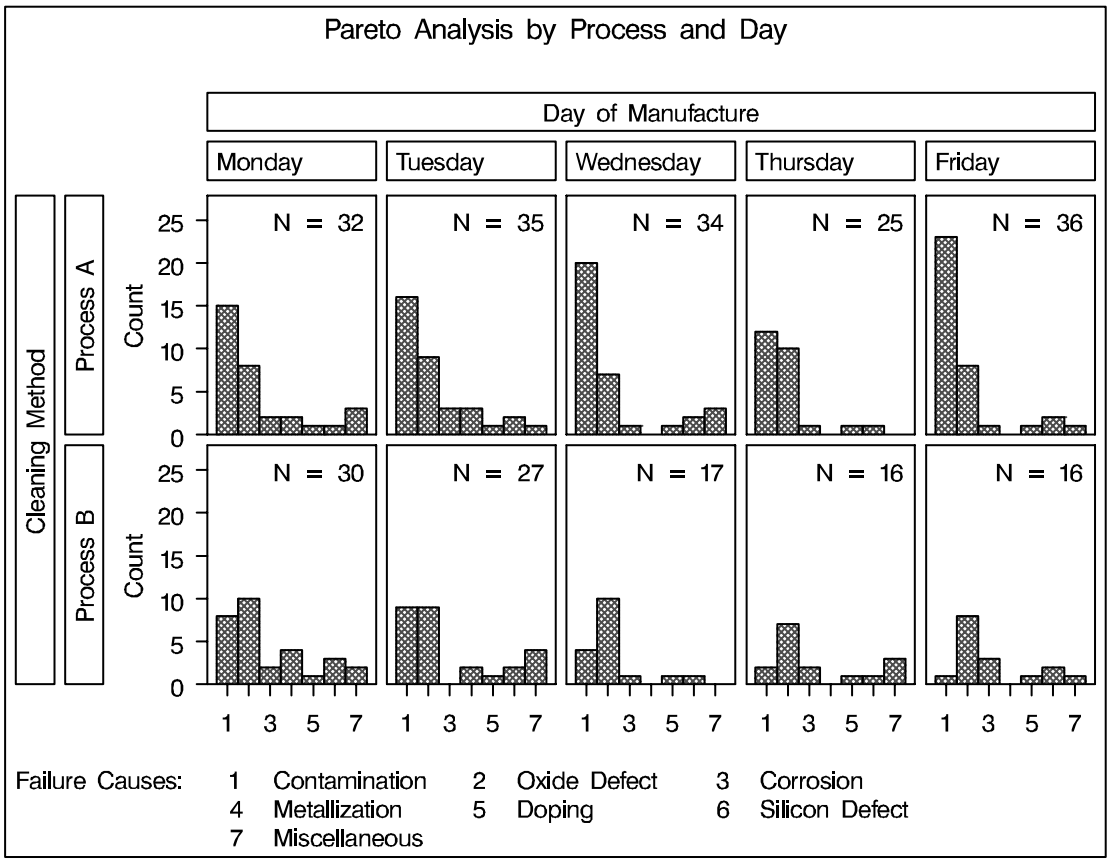
```
title 'Pareto Analysis by Process and Day' ;
proc pareto data=failure7;
   vbar cause / class       = ( process day )
                freq        = counts
                nrows       = 2
                ncols       = 5
                cbars       = green
                pbars       = m5x45
                last        = 'Miscellaneous'
                scale       = count
                catleglabel = 'Failure Causes:'
                intertile   = 1.0
                nohlabel
                nocurve
                nlegend ;
run;
```

The chart is shown in Output 36.6.1. The levels of the classification variables are determined by their formatted values. The default order in which the rows and columns are displayed is determined by the internal values of the classification variables, and, consequently, the columns appear in the order of the days of the week.

If DAY had been defined as a character variable with values `Monday` through `Friday`, the columns in Output 36.6.1 would have appeared in alphabetical order.

You can override the default order with the ORDER1= and ORDER2= options , which are described on pages 990–991.

Output 36.6.1. Controlling Row and Column Order



Example 36.7. Merging Columns in a Comparative Pareto Chart

This example is a continuation of Example 36.4 and illustrates a method for merging the columns in a comparative Pareto chart.

See PARETO14 in the SAS/QC Sample Library

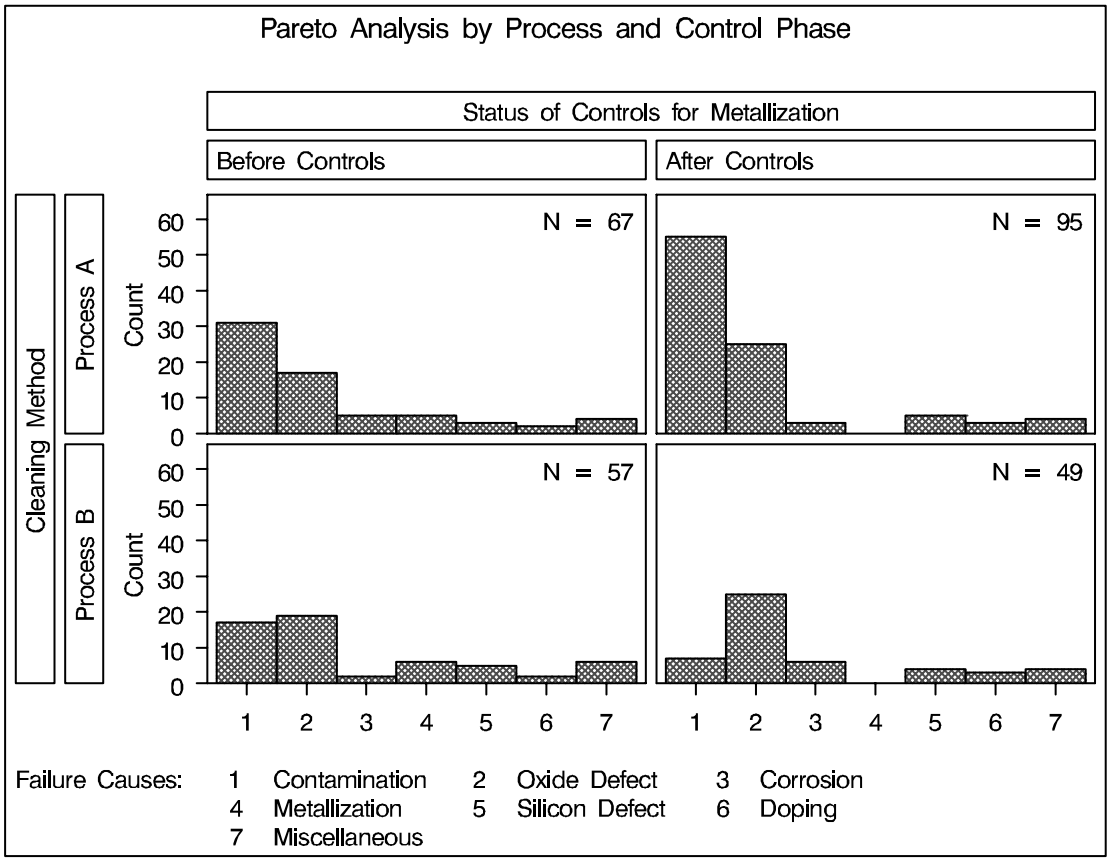
Suppose that controls for metallization were introduced on Wednesday. To show the effect of the controls, the columns for *Monday* and *Tuesday* are to be merged into a column labeled *Before Controls*, and the remaining columns are to be merged into a column labeled *After Controls*. The following statements introduce a format named CNTLFMT that merges the levels of DAY:

```
proc format;
   value cntlfmt   1-2 = 'Before Controls'
                   3-5 = 'After Controls' ;
```

The following statements create the chart shown in Output 36.7.1:

```
title 'Pareto Analysis by Process and Control Phase' ;
proc pareto data=failure7;
   vbar cause / class       = ( process day )
                freq        = counts
                cbars       = green
                pbars       = m5x45
                last        = 'Miscellaneous'
                scale       = count
                catleglabel = 'Failure Causes:'
                intertile   = 1.0
                nohlabel
                nocurve
                nlegend ;
   format day cntlfmt. ;
   label  day = 'Status of Controls for Metallization';
run;
```

Output 36.7.1. Merging Classification Levels



The levels of DAY are determined by its formatted values, `Before Controls` and `After Controls`. By default, the order in which the columns are displayed is

determined by the internal values. In this example, there are multiple distinct internal values for each level, and the procedure uses the internal value that occurs first in the input data set.

Example 36.8. Creating Weighted Pareto Charts

See PARETO12 in the SAS/QC Sample Library

In many applications, you can quantify the priority or severity of a problem with a measure such as the cost of repair or the loss to the customer expressed in man-hours. This example shows how to analyze such data with a weighted Pareto chart that incorporates the cost.

Suppose that the cost associated with each of the problems in data set FAILURE7 (see Example 36.6 on page 1071) has been determined and that the costs have been converted to a relative scale. The following statements add the cost information to the data set:

```
data failure7;
   length analysis $ 16 ;
   label analysis = 'Basis for Analysis' ;
   set failure7;
   analysis = 'Cost' ;
      if      cause = 'Contamination'  then cost = 3.0 ;
      else if cause = 'Metallization'  then cost = 8.5 ;
      else if cause = 'Oxide Defect'   then cost = 9.5 ;
      else if cause = 'Corrosion'      then cost = 2.5 ;
      else if cause = 'Doping'         then cost = 3.6 ;
      else if cause = 'Silicon Defect' then cost = 3.4 ;
      else                                  cost = 1.0 ;
      output;
   analysis = 'Frequency' ;
      cost = 1.0 ;
      output;
run;
```

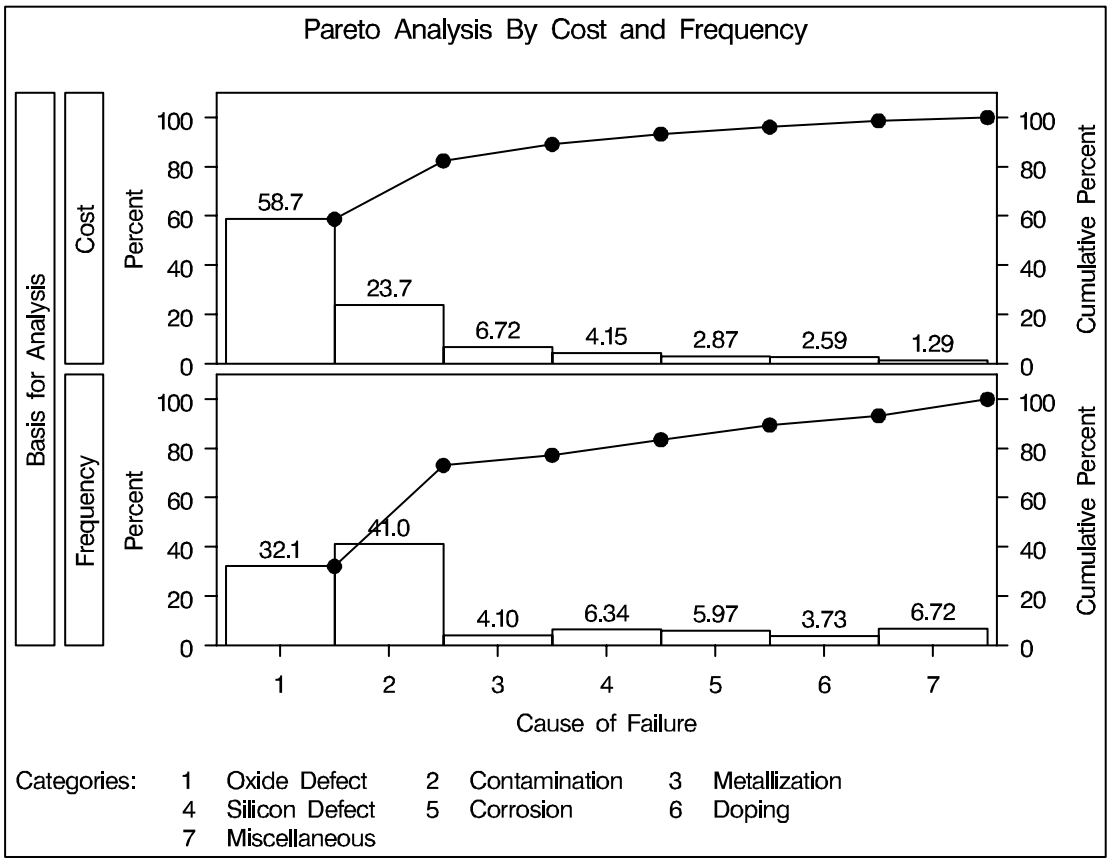
The classification variable ANALYSIS has two levels, `Cost` and `Frequency`. For ANALYSIS=`Cost`, the value of COST is the relative cost, and for ANALYSIS=`Frequency`, the value of COST is one.

The following statements create a one-way comparative Pareto chart with ANALYSIS as the classification variable, in which the cells are weighted Pareto charts with COST as the weight variable:

```
title 'Pareto Analysis By Cost and Frequency' ;
proc pareto data=failure7;
   vbar cause / class      = ( analysis )
                freq       = counts
                weight     = cost
                barlabel   = value
                out        = summary
                intertile  = 1.0 ;
run;
```

The display is shown in Output 36.8.1.

Output 36.8.1. Taking Cost into Account



Within each cell, the height of a bar is the frequency of the category multiplied by the value of COST, expressed as a percent of the total across all categories. Thus, for the cell in which ANALYSIS is equal to `Frequency`, the bars simply indicate the frequencies expressed in percent units. This display shows that the most commonly occurring problem (Contamination) is not the most expensive problem (Oxide Defect). The output data set SUMMARY is listed in Output 36.8.2.

Output 36.8.2. The Output Data Set SUMMARY

Obs	analysis	cause	cost	_COUNT_	_WCOUNT_	_PCT_	_CMPCT_
1	Cost	Oxide Defect	9.5	86	817.0	58.6799	58.680
2	Cost	Contamination	3.0	110	330.0	23.7018	82.382
3	Cost	Metallization	8.5	11	93.5	6.7155	89.097
4	Cost	Silicon Defect	3.4	17	57.8	4.1514	93.249
5	Cost	Corrosion	2.5	16	40.0	2.8729	96.122
6	Cost	Doping	3.6	10	36.0	2.5856	98.707
7	Cost	Miscellaneous	1.0	18	18.0	1.2928	100.000
8	Frequency	Oxide Defect	1.0	86	86.0	32.0896	32.090
9	Frequency	Contamination	1.0	110	110.0	41.0448	73.134
10	Frequency	Metallization	1.0	11	11.0	4.1045	77.239
11	Frequency	Silicon Defect	1.0	17	17.0	6.3433	83.582
12	Frequency	Corrosion	1.0	16	16.0	5.9701	89.552
13	Frequency	Doping	1.0	10	10.0	3.7313	93.284
14	Frequency	Miscellaneous	1.0	18	18.0	6.7164	100.000

References

Burr, J. T. (1990), "The Tools of Quality, Part VI: Pareto Charts," *Quality Progress*, 23 (11), 59–61.

Cleveland, W. S. (1985), *The Elements of Graphing Data*, Monterey, California: Wadsworth, Inc.

Ishikawa, K. (1976), *Guide To Quality Control*, Tokyo: Asian Productivity Organization.

Kume, H. (1985), *Statistical Methods for Quality Improvement*, Tokyo: AOTS Chosakai, Ltd.

SAS Institute Inc. (1999), *SAS/GRAPH Software: Reference, Version 8*, Cary, NC: SAS Institute Inc.

Wadsworth, H. M., Stephens, K. S., and Godfrey, A. B. (1986), *Modern Methods for Quality Control and Improvement*, New York: John Wiley & Sons, Inc.

Part 9
The RELIABILITY Procedure

Contents

Chapter 37. The RELIABILITY Procedure 1081

References 1217

Chapter 37
The RELIABILITY Procedure

Chapter Contents

OVERVIEW 1083

GETTING STARTED 1085
- Analysis of Right-Censored Data from a Single Population 1085
- Weibull Analysis Comparing Groups of Data 1088
- Analysis of Accelerated Life Test Data 1090
- Weibull Analysis of Interval Data with Common Inspection Schedule 1096
- Lognormal Analysis with Arbitrary Censoring 1100
- Regression Modeling 1104
- Regression Model with Non-Constant Scale 1109
- Regression Model with Two Independent Variables 1112
- Weibull Probability Plot for Two Combined Failure Modes 1115
- Analysis of Recurrence Data on Repairs 1118
- Comparison of Two Samples of Repair Data 1122
- Analysis of Interval Age Recurrence Data 1126
- Analysis of Binomial Data 1129

SYNTAX 1132
- Primary Statements 1132
- Secondary Statements 1132
- Graphical Enhancement Statements 1133
- PROC RELIABILITY Statement 1134
- ANALYZE Statement 1134
- BY Statement 1138
- CLASS Statement 1138
- DISTRIBUTION Statement 1138
- FMODE Statement 1139
- FREQ Statement 1140
- INSET Statement 1141
- LOGSCALE Statement 1144
- MAKE Statement 1144
- MCFPLOT Statement 1145
- MODEL Statement 1152
- NENTER Statement 1156
- PROBPLOT Statement 1156
- RELATIONPLOT Statement 1164

UNITID Statement 1173

DETAILS 1174
Abbreviations and Notation 1174
Types of Lifetime Data 1174
Probability Distributions 1174
Probability Plotting 1177
Nonparametric Confidence Intervals for Cumulative Failure Probabilities . . 1186
Parameter Estimation 1188
Regression Model Observation-Wise Statistics 1203
Recurrence Data from Repairable Systems 1208
ODS Table Names 1213

Chapter 37
The RELIABILITY Procedure

Overview

The RELIABILITY procedure provides tools for reliability and survival data analysis and for recurrence data analysis. You can use this procedure to

- construct probability plots and fitted life distributions with left-, right-, and interval-censored lifetime data
- fit regression models, including accelerated life test models, to combinations of left-, right-, and interval-censored data
- analyze recurrence data from repairable systems

These tools benefit reliability engineers and industrial statisticians working with product life data and system repair data. They also aid workers in other fields, such as medical research, pharmaceuticals, social sciences, and business, where survival and recurrence data are analyzed.

Most practical problems in reliability data analysis involve right-censored, left-censored, or interval-censored data. The RELIABILITY procedure provides probability plots of uncensored, right-censored, interval-censored data, and arbitrarily censored data.

Features of the RELIABILITY procedure include

- probability plotting and parameter estimation for the common life distributions: Weibull, exponential, extreme value, normal, lognormal, logistic, and loglogistic. The data can be complete, right censored, or interval censored.
- maximum likelihood estimates of distribution parameters, percentiles, and reliability functions
- both asymptotic normal and likelihood ratio confidence intervals for distribution parameters and percentiles. Asymptotic normal confidence intervals for the reliability function are also available.
- estimation of distribution parameters by least squares fitting to the probability plot
- Weibayes analysis, where there are no failures and where the data analyst specifies a value for the Weibull shape parameter
- estimates of the resulting distribution when specified failure modes are eliminated
- plots of the data and the fitted relation for life versus stress in the analysis of accelerated life test data

- fitting of regression models to life data, where the life distribution location parameter is a linear function of covariates. The fitting yields maximum likelihood estimates of parameters of a regression model with a Weibull, exponential, extreme value, normal, lognormal, logistic and loglogistic, or generalized gamma distribution. The data can be complete, right censored, left censored, or interval censored. For example, accelerated life test data can be modeled with such a regression model.
- nonparametric estimates and plots of the mean cumulative function for cost or number of recurrences and associated confidence intervals from data with exact or interval recurrence ages.

Some of the features provided in the RELIABILITY procedure are available in other SAS procedures.

- You can construct probability plots of life data with the CAPABILITY procedure; however, the CAPABILITY procedure is intended for process capability analysis rather than reliability analysis, and the data must be complete, that is, uncensored.
- The LIFEREG procedure fits regression models with life distributions such as the Weibull, lognormal, and loglogistic to left-, right-, and interval-censored data. The RELIABILITY procedure fits the same distributions and regression models as the LIFEREG procedure and, in addition, provides a graphical display of life data in probability plots.

The books by Lawless (1982), Meeker and Escobar (1998), Nelson (1982), Nelson (1990), and Tobias and Trindade (1995) provide many examples taken from diverse fields and describe the analyses provided by the RELIABILITY procedure. Nelson emphasizes reliability data analysis from an engineering viewpoint.

The features of the procedure that deal with the analysis of repair data from systems are based on the work of Doganaksoy and Nelson (1991), Nelson (1988), Nelson (1995), Nelson (2002), and Nelson and Doganaksoy (1989), who provide examples of repair data analysis.

Getting Started

This section introduces the RELIABILITY procedure with examples that illustrate some of the analyses that it performs.

Analysis of Right-Censored Data from a Single Population

The Weibull distribution is used in a wide variety of reliability analysis applications. This example illustrates the use of the Weibull distribution to model product life data from a single population. The following statements create a SAS data set containing observed and right-censored lifetimes of 70 diesel engine fans (Nelson 1982, p. 318).

```
data fan;
   input lifetime censor @@;
   lifetime = lifetime/1000;
   label lifetime='Fan Life (1000s of Hours)';
   datalines;
    450 0    460 1   1150 0   1150 0   1560 1
   1600 0   1660 1   1850 1   1850 1   1850 1
   1850 1   1850 1   2030 1   2030 1   2030 1
   2070 0   2070 0   2080 0   2200 1   3000 1
   3000 1   3000 1   3000 1   3100 0   3200 1
   3450 0   3750 1   3750 1   4150 1   4150 1
   4150 1   4150 1   4300 1   4300 1   4300 1
   4300 1   4600 0   4850 1   4850 1   4850 1
   4850 1   5000 1   5000 1   5000 1   6100 1
   6100 0   6100 1   6100 1   6300 1   6450 1
   6450 1   6700 1   7450 1   7800 1   7800 1
   8100 1   8100 1   8200 1   8500 1   8500 1
   8500 1   8750 1   8750 0   8750 1   9400 1
   9900 1  10100 1  10100 1  10100 1  11500 1
   ;
run;
```

Some of the fans had not failed at the time the data were collected, and the unfailed units have right-censored lifetimes. The variable Lifetime represents either a failure time or a censoring time in thousands of hours. The variable Censor is equal to 0 if the value of Lifetime is a failure time, and it is equal to 1 if the value is a censoring time.

The following statements use the RELIABILITY procedure to produce the graphical output shown in Figure 37.1:

```
proc reliability data=fan;
   distribution weibull;
   pplot lifetime*censor( 1 ) /  covb ;
run;
```

The DISTRIBUTION statement specifies the Weibull distribution for probability plotting and maximum likelihood (ML) parameter estimation. The PROBPLOT statement produces a probability plot for the variable Lifetime and specifies that the value

of 1 for the variable Censor denotes censored observations. You can specify any value, or group of values, for the *censor-variable* (in this case, Censor) to indicate censoring times. The option COVB requests the ML parameter estimate covariance matrix.

The graphical output, displayed in Figure 37.1, consists of a probability plot of the data, an ML fitted distribution line, and confidence intervals for the percentile (life-time) values. An *inset* box containing summary statistics, Weibull scale and shape estimates, and other information is displayed on the plot by default. The locations of the right-censored data values are plotted in an area at the top of the plot.

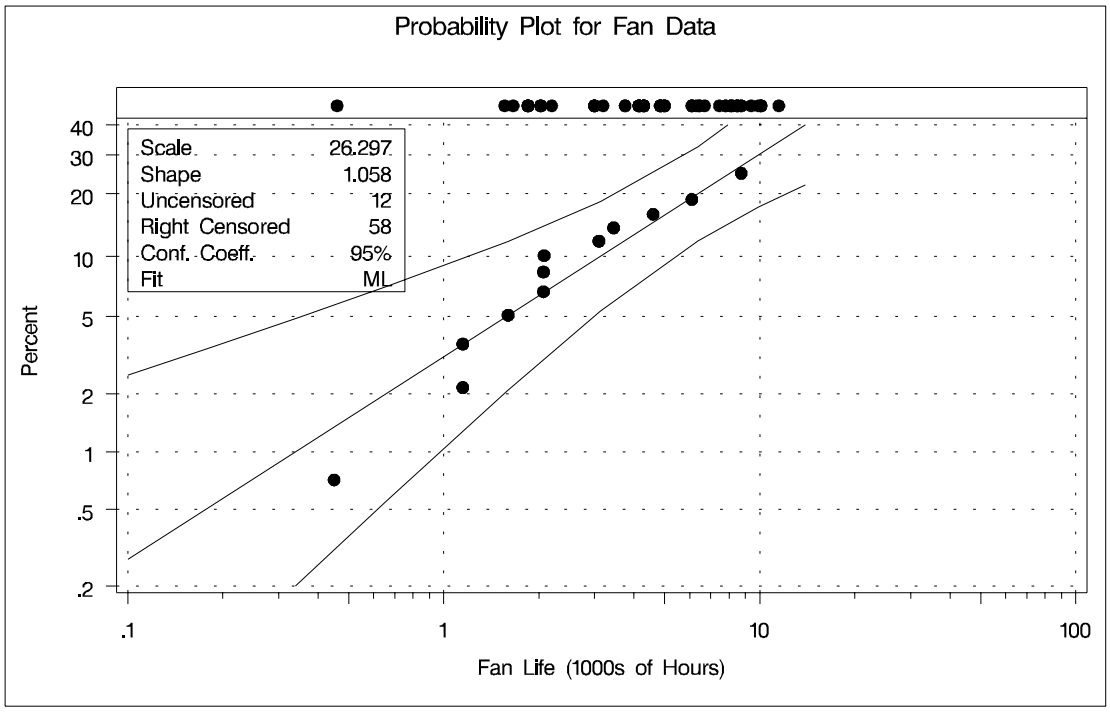


Figure 37.1. Weibull Probability Plot for the Engine Fan Data

The tabular output produced by the preceding SAS statements is shown in Figure 37.2 and Figure 37.3. This consists of summary data, fit information, parameter estimates, distribution percentile estimates, standard errors, and confidence intervals for all estimated quantities.

The RELIABILITY Procedure

Model Information

Input Data Set	WORK.FAN	
Analysis Variable	lifetime	Fan Life (1000s of Hours)
Censor Variable	censor	
Distribution	Weibull	
Estimation Method	Maximum Likelihood	
Confidence Coefficient	95%	
Observations Used	70	

Algorithm converged.

Summary of Fit

Observations Used	70
Uncensored Values	12
Right Censored Values	58
Maximum Loglikelihood	-42.248

Weibull Parameter Estimates

Parameter	Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits Lower	Upper
EV Location	3.2694	0.4659	2.3563	4.1826
EV Scale	0.9448	0.2394	0.5749	1.5526
Weibull Scale	26.2968	12.2514	10.5521	65.5344
Weibull Shape	1.0584	0.2683	0.6441	1.7394

Other Weibull Distribution Parameters

Parameter	Value
Mean	25.7156
Mode	1.7039
Median	18.6002
Standard Deviation	24.3066

Estimated Covariance Matrix Weibull Parameters

	EV Location	EV Scale
EV Location	0.21705	0.09044
EV Scale	0.09044	0.05733

Estimated Covariance Matrix Weibull Parameters

	Weibull Scale	Weibull Shape
Weibull Scale	150.09724	-2.66446
Weibull Shape	-2.66446	0.07196

Figure 37.2. Tabular Output for the Fan Data Analysis

Weibull Percentile Estimates

Percent	Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits Lower	Upper
0.1	0.03852697	0.05027782	0.002985	0.49726229
0.2	0.07419554	0.08481353	0.00789519	0.69725757
0.5	0.17658807	0.16443381	0.02846732	1.09540855
1	0.34072273	0.2635302	0.07482449	1.55152389
2	0.65900116	0.40845639	0.19556981	2.22060107
5	1.58925244	0.68465855	0.68311002	3.69738878
10	3.13724079	0.99379006	1.68620756	5.83693255
20	6.37467675	1.74261908	3.73051433	10.8930029
30	9.92885165	3.00353842	5.48788931	17.9635721
40	13.9407124	4.85766683	7.04177638	27.5986417
50	18.6002319	7.40416922	8.52475116	40.5840149
60	24.2121441	10.8733301	10.0408557	58.3842593
70	31.3378076	15.750336	11.7018888	83.9230489
80	41.2254517	23.1787018	13.6956839	124.092954
90	57.8253251	36.9266698	16.5405275	202.156081
95	74.1471722	51.6127806	18.9489625	290.137423
99	111.307797	88.1380261	23.5781482	525.462197
99.9	163.265082	144.264145	28.8905203	922.637827

Figure 37.3. Percentile Estimates for the Fan Data Analysis

Weibull Analysis Comparing Groups of Data

This example illustrates probability plotting and distribution fitting for data grouped by the levels of a special *group-variable*. The data are from an accelerated life test of an insulating fluid and are the times to electrical breakdown of the fluid under different high voltage levels. Each voltage level defines a subset of data for which a separate analysis and Weibull plot are produced. These data are the 26kV, 30kV, 34kV, and 38kV groups of the data provided by Nelson (1990, p. 129). The following statements create a SAS data set containing the lifetimes and voltages.

```
data fluid;
    input time voltage$ @@;
    datalines;
    5.79    26kv    1579.52 26kv
    2323.7  26kv    7.74    30kv
    17.05   30kv    20.46   30kv
    21.02   30kv    22.66   30kv
    43.4    30kv    47.3    30kv
    139.07  30kv    144.12  30kv
    175.88  30kv    194.90  30kv
    .19     34kv    .78     34kv
    .96     34kv    1.31    34kv
    2.78    34kv    3.16    34kv
    4.15    34kv    4.67    34kv
    4.85    34kv    6.50    34kv
    7.35    34kv    8.01    34kv
    8.27    34kv    12.06   34kv
    31.75   34kv    32.52   34kv
    33.91   34kv    36.71   34kv
    72.89   34kv    .09     38kv
    .39     38kv    .47     38kv
```

```
      .73      38kv     .74      38kv
      1.13     38kv     1.40     38kv
      2.38     38kv
      ;
run;
```

The variable Time provides the time to breakdown in minutes, and the variable Voltage provides the voltage level at which the test was conducted. These data are not censored.

The RELIABILITY procedure plots the data for the different voltage levels on the same Weibull probability plot, fits a separate distribution to the data at each voltage level, and superimposes distribution lines on the plot.

The following statements produce the probability plot shown in Figure 37.4 for the variable Time at each level of the *group-variable* Voltage.

```
proc reliability data=fluid;
   distribution weibull;
   pplot time=voltage  / overlay
                         noconf;
run;
```

The input data set FLUID is specified by the DATA= option in the PROC RELIABILITY statement. The PROBPLOT statement option OVERLAY specifies that plots for the groups are to be overlaid rather than displayed separately. The option NOCONF specifies that no confidence bands are to be plotted, since these can interfere with one another on overlaid plots; confidence bands are displayed by default.

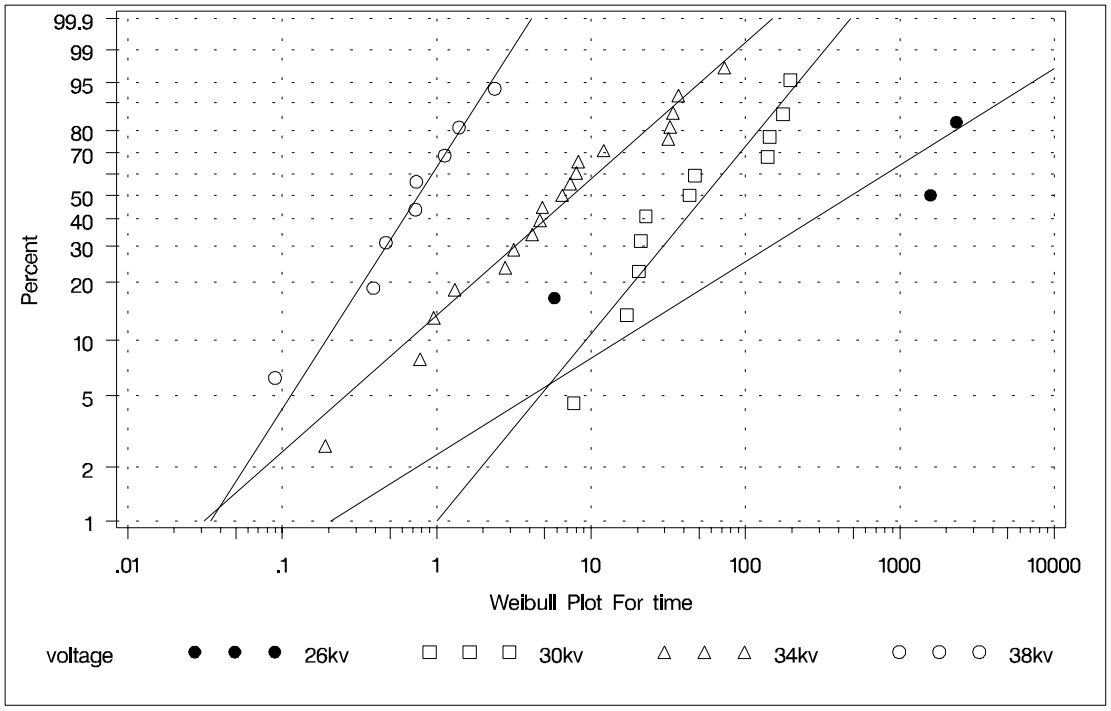


Figure 37.4. Weibull Probability Plot for the Insulating Fluid Data

A summary table that contains information for all groups is displayed. In addition, information identical to that shown in Figure 37.2 is tabulated for each level of voltage. The summary table for all groups and the tables for the 26kV group are shown in Figure 37.5 and Figure 37.6.

Model Information - All Groups

Input Data Set	WORK.FLUID
Analysis Variable	time
Distribution	Weibull
Estimation Method	Maximum Likelihood
Confidence Coefficient	95%
Observations Used	41

Algorithm converged for group 26kv.

Summary of Fit

		Group
Observations Used	3	26kv
Uncensored Values	3	26kv
Maximum Loglikelihood	-6.845551	26kv

Figure 37.5. Partial Listing of the Tabular Output for the Insulating Fluid Data

Analysis of Accelerated Life Test Data

The following example illustrates the analysis of an accelerated life test for Class-B electrical motor insulation using data provided by Nelson (1990, p. 243). Forty insulation specimens were tested at four temperatures: 150°, 170°, 190°, and 220°C. The purpose of the test is to estimate the median life of the insulation at the design operating temperature of 130°C.

The following SAS program creates the data listed in Figure 37.7. Ten specimens of the insulation were tested at each test temperature. The variable Time provides a specimen time to failure or a censoring time, in hours. The variable Censor is equal to 1 if the value of the variable Time is a right-censoring time and is equal to 0 if the value is a failure time. Some censor times and failure times are identical at some of the temperatures. Rather than repeating identical observations in the input data set, the variable Count provides the number of specimens with identical times and temperatures. The variable Temp provides the test temperature in degrees centigrade. The variable Cntrl is a control variable specifying that percentiles are to be computed only for the first value of Temp (130°C). The value of Temp in the first observation (130°C) does not correspond to a test temperature. The missing values in the first observation cause the observation to be excluded from the model fit, and the value of 1 for the variable Cntrl causes percentiles corresponding to a temperature of 130°C to be computed.

Model Information - All Groups

Input Data Set	WORK.FLUID
Analysis Variable	time
Distribution	Weibull
Estimation Method	Maximum Likelihood
Confidence Coefficient	95%
Observations Used	41

Weibull Parameter Estimates

Parameter	Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits Lower	Upper	Group
EV Location	6.8625	1.1040	4.6986	9.0264	26kv
EV Scale	1.8342	0.9611	0.6568	5.1226	26kv
Weibull Scale	955.7467	1055.1862	109.7941	8319.6794	26kv
Weibull Shape	0.5452	0.2857	0.1952	1.5226	26kv

Other Weibull Distribution Parameters

Parameter	Value	Group
Mean	1649.4882	26kv
Mode	0.0000	26kv
Median	487.9547	26kv
Standard Deviation	3279.0212	26kv

Weibull Percentile Estimates

Percent	Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits Lower	Upper	Group
0.1	0.00300636	0.02113841	3.11203E-9	2904.27046	26kv
0.2	0.01072998	0.06838144	4.03597E-8	2852.65767	26kv
0.5	0.0577713	0.31803193	1.19079E-6	2802.78862	26kv
1	0.20695478	1.00385021	0.00001538	2784.16263	26kv
2	0.74484901	3.12705686	0.00019885	2790.0941	26kv
5	4.1142692	13.7388263	0.00591379	2862.3304	26kv
10	15.406565	41.4763373	0.07873508	3014.69497	26kv
20	61.0231127	125.020566	1.10053199	3383.65475	26kv
30	144.246801	242.203982	5.36856883	3875.73303	26kv
40	278.770459	398.048692	16.9761581	4577.77125	26kv
50	487.954708	610.02855	42.0948552	5656.26835	26kv
60	814.147288	920.537706	88.770543	7466.84412	26kv
70	1343.42243	1433.97868	165.818889	10884.0666	26kv
80	2287.87124	2445.52431	281.5628	18590.3635	26kv
90	4412.96962	5148.34986	448.419608	43428.7452	26kv
95	7150.89745	9248.2654	566.892142	90202.9338	26kv
99	15735.8513	24666.0388	728.831025	339745.437	26kv
99.9	33104.172	62018.1074	841.826189	1301796.28	26kv

Figure 37.6. Partial Listing of the Tabular Output for the Insulating Fluid Data

```
data classb;
   input hours temp count censor;
   if _n_ = 1 then cntrl=1;
   else cntrl=0;
   label hours='Hours';
   datalines;
      . 130 .  .
   8064 150 10 1
   1764 170 1  0
   2772 170 1  0
   3444 170 1  0
   3542 170 1  0
   3780 170 1  0
   4860 170 1  0
   5196 170 1  0
   5448 170 3  1
    408 190 2  0
   1344 190 2  0
   1440 190 1  0
   1680 190 5  1
    408 220 2  0
    504 220 3  0
    528 220 5  1
   ;
run;
```

Obs	hours	temp	count	censor	cntrl
1	.	130	.	.	1
2	8064	150	10	1	0
3	1764	170	1	0	0
4	2772	170	1	0	0
5	3444	170	1	0	0
6	3542	170	1	0	0
7	3780	170	1	0	0
8	4860	170	1	0	0
9	5196	170	1	0	0
10	5448	170	3	1	0
11	408	190	2	0	0
12	1344	190	2	0	0
13	1440	190	1	0	0
14	1680	190	5	1	0
15	408	220	2	0	0
16	504	220	3	0	0
17	528	220	5	1	0

Figure 37.7. Listing of the Class B Insulation Data

An Arrhenius-lognormal model is fitted to the data in this example. In other words, the fitted model has the lognormal (base 10) distribution, and its location parameter μ depends on the centigrade temperature Temp through the Arrhenius relationship

$$\mu(x) = \beta_0 + \beta_1 x$$

where

$$x = \frac{1000}{\text{Temp} + 273.15}$$

is 1000 times the reciprocal absolute temperature. The lognormal (base e) distribution is also available.

The following SAS statements fit the Arrhenius-lognormal model, and they display the fitted model distributions side-by-side on the probability and the relation plots shown in Figure 37.8.

```
proc reliability;
   distribution lognormal10;
   freq count;
   model hours*censor(1) = temp /
      relation=arr
      obstats( q=.1 .5 .9 control=cntrl );
   rplot hours*censor(1) = temp /
      pplot
      fit=model
      noconf
      relation = arr
      plotdata
      plotfit 10 50 90
      lupper = 1.e5
      slower=120;
   run;
```

The PROC RELIABILITY statement invokes the procedure and specifies CLASSB as the input data set. The DISTRIBUTION statement specifies that the lognormal (base 10) distribution is to be used for maximum likelihood parameter estimation and probability plotting. The FREQ statement specifies that the variable Count is to be used as a frequency variable; that is, if Count=n, then there are n specimens with the time and temperature specified in the observation.

The MODEL statement fits a linear regression equation for the distribution location parameter as a function of independent variables. In this case, the MODEL statement also transforms the independent variable through the Arrhenius relationship. The dependent variable is specified as Time. A value of 1 for the variable Censor indicates that the corresponding value of Time is a right-censored observation; otherwise, the value is a failure time. The temperature variable Temp is specified as the independent variable in the model. The MODEL statement option RELATION=ARR specifies the Arrhenius relationship.

The option OBSTATS requests observation-wise statistics. The options in parentheses following OBSTATS indicate which statistics are to be computed. In this case, QUANTILE= .1 .5 .9 specifies that quantiles of the fitted distribution are to be computed for the value of the variable Temp at each observation. The CONTROL= option requests quantiles only for those observations in which the variable Cntrl has a value of 1. This eliminates unnecessary quantiles in the OBSTATS table since, in this case, only the quantiles at the design temperature of 130°C are of interest.

The RPLOT, or RELATIONPLOT, statement displays a plot of the lifetime data and the fitted model. The dependent variable Time, the independent variable Temp, and the censoring indicator Censor are the same as in the MODEL statement. The option FIT=MODEL specifies that the model fitted with the preceding MODEL state-

ment is to be used for probability plotting and in the relation plot. The option RELATION=ARR specifies an Arrhenius scale for the horizontal axis of the relation plot. The PPLOT option specifies that a probability plot is to be displayed alongside the relation plot. The type of probability plot is determined by the distribution named in the DISTRIBUTION statement, in this case, a lognormal (base 10) distribution. Weibull, extreme value, lognormal (base e), normal, loglogistic, and logistic distributions are also available. The NOCONF option suppresses the default percentile confidence bands on the probability plot. The PLOTDATA option specifies that the failure times are to be plotted on the relation plot. The PLOTFIT option specifies that the 10th, 50th, and 90th percentiles of the fitted relationship are to be plotted on the relation plot. The options LUPPER and SLOWER specify an upper limit on the life axis scale and a lower limit on the stress (temperature) axis scale in the plots.

The plots produced by the preceding statements are shown in Figure 37.8. The plot on the left is an overlaid lognormal probability plot of the data and the fitted model. The plot on the right is a relation plot showing the data and the fitted relation. The fitted straight lines are percentiles of the fitted distribution at each temperature. An Arrhenius relation fitted to the data, plotted on an Arrhenius plot, yields straight percentile lines.

Since all the data at $150°$C are right censored, there are no failures corresponding to $150°$C on the probability plot. However, the fitted distribution at $150°$C is plotted on the probability plot.

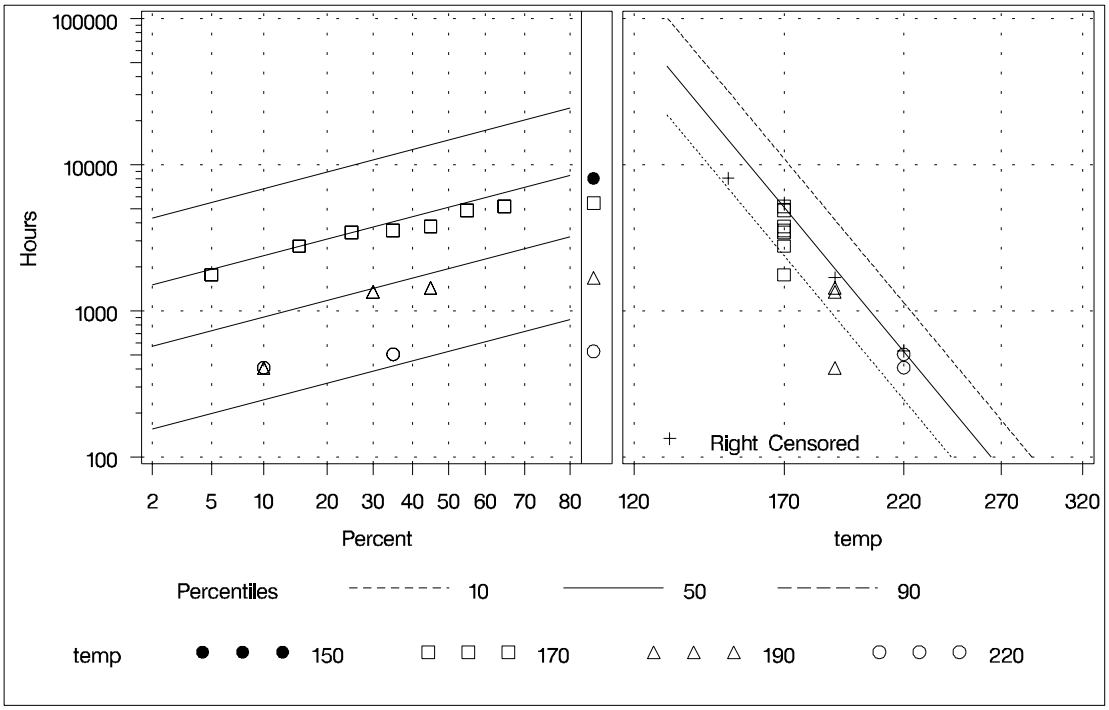


Figure 37.8. Probability and Relation Plots for the Class B Insulation Data

The tabular output requested with the MODEL statement is shown in Figure 37.9. The "Model Information" table provides general information about the data and

model. The "Summary of Fit" table shows the number of observations used, the number of failures and of censored values (accounting for the frequency count), and the maximum log likelihood for the fitted model.

The "Lognormal Parameter Estimates" table contains the Arrhenius-lognormal model parameter estimates, their standard errors, and confidence interval estimates. In this table, INTERCEPT is the maximum likelihood estimate of β_0, TEMP is the estimate of β_1, and Scale is the estimate of the lognormal scale parameter, σ.

Model Information

Input Data Set	WORK.CLASSB	
Analysis Variable	hours	Hours
Relation	Arrhenius(temp)	
Censor Variable	censor	
Frequency Variable	count	
Distribution	Lognormal (Base 10)	

Algorithm converged.

Summary of Fit

Observations Used	16
Uncensored Values	17
Right Censored Values	23
Missing Observations	1
Maximum Loglikelihood	-12.96533

Lognormal Parameter Estimates

Parameter	Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits Lower	Upper
Intercept	-6.0182	0.9467	-7.8737	-4.1628
temp	4.3103	0.4366	3.4546	5.1660
Scale	0.2592	0.0473	0.1812	0.3708

Observation Statistics

Hours	censor	temp	count	Prob	Pcntl
.	.	130	.	0.1000	21937.658
.	.	130	.	0.5000	47135.132
.	.	130	.	0.9000	101274.29

Observation Statistics

Hours	Stderr	Lower	Upper
.	6959.151	11780.636	40851.857
.	16125.548	24106.685	92162.016
.	42061.1	44872.401	228569.92

Figure 37.9. MODEL Statement Output for the Class B Data

The "Observation Statistics" table provides the estimates of the fitted distribution quantiles, their standard errors, and confidence limits. These are given only for the value of 130°C, as specified with the CONTROL= option in the MODEL statement. The predicted median life at 130°C corresponds to a quantile of 0.5, and it is approx-

imately 47,134 hours.

In addition to the MODEL statement output in Figure 37.9, the RELIABILITY procedure produces tabular output for each temperature that is identical to the output produced with the PROBPLOT statement. This output is not shown here.

Weibull Analysis of Interval Data with Common Inspection Schedule

Table 37.1 shows data for 167 identical turbine parts provided by Nelson (1982, p. 415). The parts were inspected at certain times to determine which parts had cracked since the last inspection. The times at which parts develop cracks are to be fitted with a Weibull distribution.

Table 37.1. Turbine Part Cracking Data

Inspection (Months)		Number	
Start	End	Cracked	Cumulative
0	6.12	5	5
6.12	19.92	16	21
19.92	29.64	12	33
29.64	35.40	18	51
35.40	39.72	18	69
39.72	45.24	2	71
45.24	52.32	6	77
52.32	63.48	17	94
63.48	Survived	73	167

Table 37.1 shows the time in months of each inspection period and the number of cracked parts found in each period. These data are said to be interval censored since only the time interval in which failures occurred is known, not the exact failure times. Seventy-three parts had not cracked at the last inspection, which took place at 63.48 months. These 73 lifetimes are right censored, since the lifetimes are known only to be greater than 63.48 months.

The interval data in this example is read from a SAS data set with a special structure. All units must have a common inspection schedule. This type of interval data is called readout data. The following SAS program creates the SAS data set named CRACKS, shown in Figure 37.10, and provides the data in Table 37.1 with this structure. The variable Time is the inspection time, that is, the upper endpoint of each interval. The variable Units is the number of unfailed units at the beginning of each interval, and the variable Fail is the number of units with cracks at the inspection time.

```
data cracks;
   input time units fail;
   cards;
   6.12  167 5
   19.92 162 16
   29.64 146 12
   35.4  134 18
   39.72 116 18
```

```
45.24 98  2
52.32 96  6
63.48 90 17
;
```

Obs	time	units	fail
1	6.12	167	5
2	19.92	162	16
3	29.64	146	12
4	35.40	134	18
5	39.72	116	18
6	45.24	98	2
7	52.32	96	6
8	63.48	90	17

Figure 37.10. Listing of the Turbine Part Cracking Data

The following statements use the RELIABILITY procedure to produce the probability plot in Figure 37.11 for the data in the data set CRACKS.

```
proc reliability data=cracks;
   freq fail;
   nenter units;
   distribution weibull;
   probplot time / readout
                   pconfplt
                   noconf;
run;
```

The FREQ statement specifies that the variable FAIL provides the number of failures in each interval. The NENTER statement specifies that the variable Units provides the number of unfailed units at the beginning of each interval. The DISTRIBUTION statement specifies that the Weibull distribution is used for parameter estimation and probability plotting. The PROBPLOT statement requests a probability plot of the data.

The PROBPLOT statement option READOUT indicates that the data in the CRACKS data set are readout (or interval) data. The option PCONFPLT specifies that confidence intervals for the cumulative probability of failure are to be plotted. The confidence intervals for the cumulative probability are based on the binomial distribution for time intervals until right censoring occurs. For time intervals after right censoring occurs, the binomial distribution is not valid, and a normal approximation is used to compute confidence intervals.

The option NOCONF suppresses the display of confidence intervals for distribution percentiles in the probability plot.

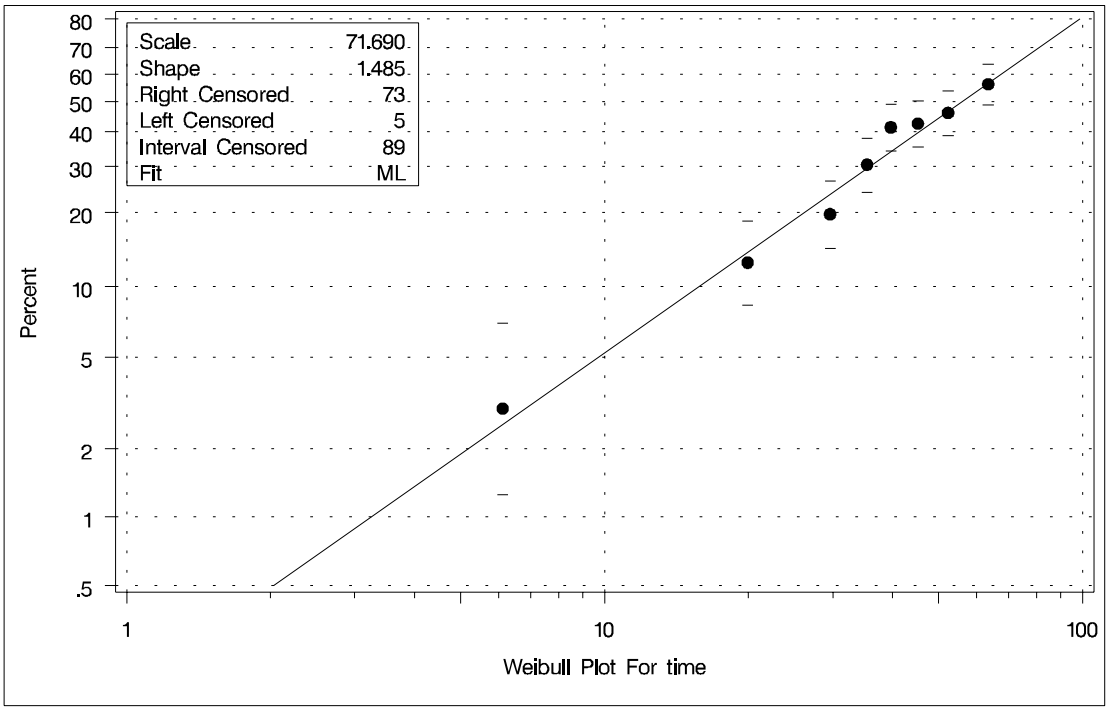


Figure 37.11. Weibull Probability Plot for the Part Cracking Data

A listing of the tabular output produced by the preceding SAS statements is shown in Figure 37.12 and Figure 37.13. By default, the specified Weibull distribution is fitted by maximum likelihood. The line plotted on the probability plot and the tabular output summarize this fit. For interval data, the estimated cumulative probabilities and associated confidence intervals are tabulated. In addition, general fit information, parameter estimates, percentile estimates, standard errors, and confidence intervals are tabulated.

Model Information

Input Data Set	WORK.CRACKS
Analysis Variable	time
Frequency Variable	fail
NENTER Variable	units
Distribution	Weibull
Estimation Method	Maximum Likelihood
Confidence Coefficient	95%
Observations Used	8

Algorithm converged.

Summary of Fit

Observations Used	8
Right Censored Values	73
Left Censored Values	5
Interval Censored Values	89
Maximum Loglikelihood	-309.6684

Figure 37.12. Partial Listing of the Tabular Output for the Part Cracking Data

Weibull Parameter Estimates

Parameter	Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits Lower	Upper
EV Location	4.2724	0.0744	4.1265	4.4182
EV Scale	0.6732	0.0664	0.5549	0.8168
Weibull Scale	71.6904	5.3335	61.9634	82.9444
Weibull Shape	1.4854	0.1465	1.2242	1.8022

Other Weibull Distribution Parameters

Parameter	Value
Mean	64.7966
Mode	33.7622
Median	56.0144
Standard Deviation	44.3943

Weibull Percentile Estimates

Percent	Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits Lower	Upper
0.1	0.68534385	0.29999861	0.29060848	1.61625083
0.2	1.09324674	0.42889777	0.50673224	2.3586193
0.5	2.02798319	0.67429625	1.05692279	3.8912169
1	3.23938972	0.93123832	1.84401909	5.69063837
2	5.18330703	1.2581604	3.22101028	8.34106988
5	9.70579945	1.78869256	6.76335893	13.9283666
10	15.7577991	2.22445157	11.9491109	20.7804776
20	26.1159906	2.6327383	21.4337103	31.821134
30	35.8126238	2.90557264	30.547517	41.9852137
40	45.6100472	3.27409792	39.6239146	52.5005271
50	56.0143651	3.89410377	48.8792027	64.1910859
60	67.5928125	4.90210777	58.6364803	77.917165
70	81.2334227	6.46932648	69.4938134	94.9562075
80	98.7644937	8.95137184	82.6900902	117.963654
90	125.694556	13.5078386	101.821995	155.164133
95	150.057755	18.2060035	118.300075	190.340791
99	200.437864	29.1957544	150.658574	266.66479
99.9	263.348102	44.7205513	188.791789	367.347666

Figure 37.13. Partial Listing of the Tabular Output for the Part Cracking Data

In this example, the number of unfailed units at the beginning of an interval minus the number failing in the interval is equal to the number of unfailed units entering the next interval. This is not always the case since some unfailed units might be removed from the test at the end of an interval; that is, they might be right censored. The special structure of the input SAS data set required for interval data enables the RELIABILITY procedure to analyze this more general case.

Lognormal Analysis with Arbitrary Censoring

This example illustrates analyzing data that have more general censoring than in the previous example. The data can be a combination of exact failure times, left censored, right censored, and interval censored data. The intervals can be overlapping, unlike in the previous example, where the interval endpoints had to be the same for all units.

Table 37.2 shows data from Nelson (1982, p. 409), analyzed by Meeker and Escobar (1998, p. 135). Each of 435 turbine wheels was inspected once to determine whether a crack had developed in the wheel or not. The inspection time (in 100s of hours), the number inspected at the time that had cracked, and the number not cracked are shown in the table. The quantity of interest is the time for a crack to develop.

Table 37.2. Turbine Wheel Cracking Data

Inspection Time (100 hours)	Number Cracked	Number Not Cracked
4	0	39
10	4	49
14	2	31
18	7	66
22	5	25
26	9	30
30	9	33
34	6	7
38	22	12
42	21	19
46	21	15

These data consist only of left and right censored lifetimes. If a unit has developed a crack at an inspection time, the unit is left-censored at the time; if a unit has not developed a crack, it is right-censored at the time. For example, there are 4 left-censored lifetimes and 49 right-censored lifetimes at 1000 hours.

The following statements create a SAS data set named TURBINE that contains the data in the format necessary for analysis by the RELIABILITY procedure.

```
data turbine;
   label t1 = 'Time of Cracking (Hours x 100 )';
   input t1 t2 f;
   datalines;
.   4  0
4   .  39
.  10  4
10  .  49
.  14  2
14  .  31
.  18  7
18  .  66
.  22  5
22  .  25
.  26  9
```

```
26  .  30
.  30  9
30  .  33
.  34  6
34  .  7
.  38  22
38  .  12
.  42  21
42  .  19
.  46  21
46  .  15
;
run;
```

The variables T1 and T2 represent the inspection times and determine whether the observation is right or left censored. If T1 is missing (.), then T2 represents a left-censoring time; if T2 is missing, T1 represents a right-censoring time. The variable F is the number of units that were found to be cracked for left-censored observations, or not cracked for right-censored observations at an inspection time.

The following statements use the RELIABILITY procedure to produce the probability plot in Figure 37.14 for the data in the data set TURBINE.

```
proc reliability data = turbine;
   distribution lognormal;
   freq f;
   pplot ( t1 t2 ) / maxitem = 5000
                     ppout;
   run;
```

The DISTRIBUTION statement specifies that a lognormal probability plot be created. The FREQ statement identifies the frequency variable F. The option MAXITEM = 5000 specifies that the iterative algorithm that computes the points on the probability plot can take a maximum of 5000 iterations. The algorithm does not converge for this data in the default 1000 iterations, so the maximum number of iterations needs to be increased for convergence. The option PPOUT specifies that a table of the cumulative probabilities plotted on the probability plot be printed, along with standard errors and confidence limits.

The tabular output for the maximum likelihood lognormal fit for this data is shown in Figure 37.15. Figure 37.14 shows the resulting lognormal probability plot with the computed cumulative probability estimates and the lognormal fit line.

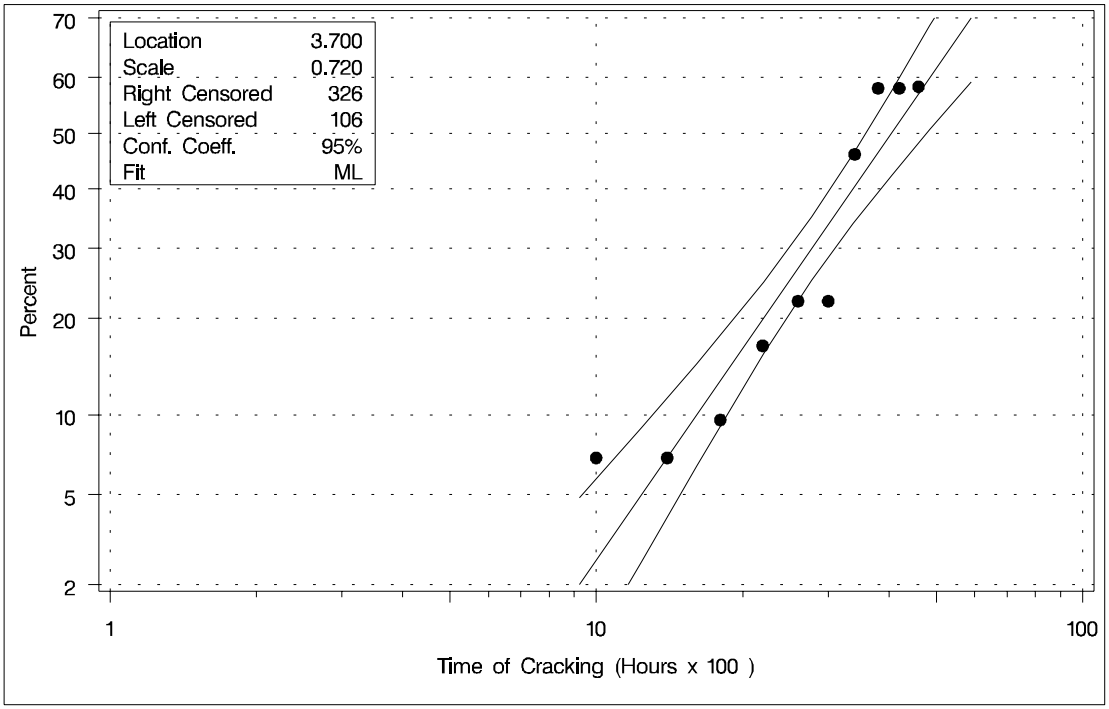


Figure 37.14. Lognormal Probability Plot for the Turbine Wheel Data

Model Information

Input Data Set	WORK.TURBINE	
Analysis Variable	t1	Time of Cracking (Hours x 100)
Analysis Variable	t2	
Frequency Variable	f	
Distribution	Lognormal (Base e)	
Estimation Method	Maximum Likelihood	
Confidence Coefficient	95%	
Observations Used	21	

Cumulative Probability Estimates

Lower Lifetime	Upper Lifetime	Cumulative Probability	Pointwise 95% Confidence Limits Lower	Upper	Standard Error
.	4	0.0000	0.0000	0.0000	0.0000
10	10	0.0698	0.0264	0.1720	0.0337
14	14	0.0698	0.0177	0.2384	0.0473
18	18	0.0959	0.0464	0.1878	0.0345
22	22	0.1667	0.0711	0.3432	0.0680
26	26	0.2222	0.1195	0.3757	0.0657
30	30	0.2222	0.1203	0.3738	0.0650
34	34	0.4615	0.2236	0.7184	0.1383
38	38	0.5809	0.4085	0.7356	0.0865
42	42	0.5809	0.4280	0.7198	0.0766
46	46	0.5836	0.4195	0.7311	0.0822

Algorithm converged.

Summary of Fit

Observations Used	21
Uncensored Values	0
Right Censored Values	326
Left Censored Values	106
Maximum Loglikelihood	-190.7315

Lognormal Parameter Estimates

Parameter	Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits Lower	Upper
Location	3.6999	0.0708	3.5611	3.8387
Scale	0.7199	0.0887	0.5655	0.9165

Other Lognormal Distribution Parameters

Parameter	Value
Mean	52.4062
Mode	24.0870
Median	40.4436
Standard Deviation	43.1855

Figure 37.15. Partial Listing of the Tabular Output for the Turbine Wheel Data

Regression Modeling

This example is an illustration of a Weibull regression model using a load accelerated life test of rolling bearings, with data provided by Nelson (1990, p. 305). Bearings are tested at four different loads, and lifetimes in 10^6 of revolutions are measured. The data are shown in Table 37.3. An outlier identified by Nelson (1990) is omitted.

Table 37.3. Bearing Lifetime Data

Load	Life (10^6 Revolutions)									
0.87	1.67	2.2	2.51	3.00	3.90	4.70	7.53	14.7	27.76	37.4
0.99	0.80	1.0	1.37	2.25	2.95	3.70	6.07	6.65	7.05	7.37
1.09	0.18	0.2	0.24	0.26	0.32	0.32	0.42	0.44	0.88	
1.18	0.073	0.098	0.117	0.135	0.175	0.262	0.270	0.350	0.386	0.456

These data are modeled with a Weibull regression model in which the independent variable is the logarithm of the load. The model is

$$\mu_i = \beta_0 + \beta_1 x_i$$

where μ_i is the location parameter of the extreme value distribution and

$$x_i = \log(\text{load})$$

for the ith bearing. The following statements create a SAS data set containing the loads, log loads, and bearing lifetimes.

```
data bearing;
   input load life @@;
   lload = log(load);
   datalines;
 .87 1.67    .87 2.2    .87 2.51    .87 3.0    .87 3.9
 .87 4.7     .87 7.53   .87 14.7    .87 27.76  .87 37.4
 .99  .8     .99 1.0    .99 1.37    .99 2.25   .99 2.95
 .99 3.7     .99 6.07   .99 6.65    .99 7.05   .99 7.37
1.09  .18   1.09  .2   1.09  .24   1.09  .26  1.09  .32
1.09  .32   1.09  .42  1.09  .44   1.09  .88  1.18  .073
1.18  .098  1.18  .117 1.18  .135  1.18  .175 1.18  .262
1.18  .270  1.18  .350 1.18  .386  1.18  .456
;
run;
```

Figure 37.16 shows a listing of the bearing data.

Obs	load	life	lload
1	0.87	1.670	-0.13926
2	0.87	2.200	-0.13926
3	0.87	2.510	-0.13926
4	0.87	3.000	-0.13926
5	0.87	3.900	-0.13926
6	0.87	4.700	-0.13926
7	0.87	7.530	-0.13926
8	0.87	14.700	-0.13926
9	0.87	27.760	-0.13926
10	0.87	37.400	-0.13926
11	0.99	0.800	-0.01005
12	0.99	1.000	-0.01005
13	0.99	1.370	-0.01005
14	0.99	2.250	-0.01005
15	0.99	2.950	-0.01005
16	0.99	3.700	-0.01005
17	0.99	6.070	-0.01005
18	0.99	6.650	-0.01005
19	0.99	7.050	-0.01005
20	0.99	7.370	-0.01005
21	1.09	0.180	0.08618
22	1.09	0.200	0.08618
23	1.09	0.240	0.08618
24	1.09	0.260	0.08618
25	1.09	0.320	0.08618
26	1.09	0.320	0.08618
27	1.09	0.420	0.08618
28	1.09	0.440	0.08618
29	1.09	0.880	0.08618
30	1.18	0.073	0.16551
31	1.18	0.098	0.16551
32	1.18	0.117	0.16551
33	1.18	0.135	0.16551
34	1.18	0.175	0.16551
35	1.18	0.262	0.16551
36	1.18	0.270	0.16551
37	1.18	0.350	0.16551
38	1.18	0.386	0.16551
39	1.18	0.456	0.16551

Figure 37.16. Listing of the Bearing Data

The following statements fit the regression model by maximum likelihood using the Weibull distribution.

```
ods output modobstats = RESIDUAL;
proc reliability data=bearing;
   distribution weibull;
   model life = lload /  covb
                         corrb
                         obstats
                         ;
run;
```

The PROC RELIABILITY statement invokes the procedure and identifies BEARING as the input data set. The DISTRIBUTION statement specifies the Weibull distribution for model fitting. The MODEL statement specifies the regression model, identi-

fying Life as the variable that provides the response values (the lifetimes) and Lload as the independent variable (the log loads). The MODEL statement option COVB requests the regression parameter covariance matrix, and the CORRB option requests the correlation matrix. The option OBSTATS requests a table that contains residuals, predicted values, and other statistics. The ODS output statement creates a SAS data set named RESIDUAL that contains the table created by the OBSTATS option.

Figure 37.17 shows the tabular output produced by the RELIABILITY procedure. The "Weibull Parameter Estimates" table contains parameter estimates, their standard errors, and 95% confidence intervals. In this table, INTERCEPT corresponds to β_0, LLOAD corresponds to β_1, and SHAPE corresponds to the Weibull shape parameter. Figure 37.18 shows a listing of the output data set RESIDUAL.

The value of the lifetime Life and the log load Lload are included in this data set, as well as statistics computed from the fitted model. The variable Xbeta is the value of the linear predictor

$$\mathbf{x}_i'\hat{\boldsymbol{\beta}} = \hat{\beta}_0 + \mathsf{Lload}\hat{\beta}_1$$

for each observation. The variable Surv contains the value of the reliability function, the variable Sresid contains the standardized residual, and the variable Aresid contains a residual adjusted for right-censored observations. Since there are no censored values in these data, Sresid is equal to Aresid for all the bearings. See Table 37.24 and Table 37.25 for other statistics that are available in the OBSTATS table and data set. See the section "Regression Model Observation-Wise Statistics" on page 1203 for a description of the residuals and other statistics.

If the fitted regression model is adequate, the standardized residuals have a standard extreme value distribution. You can check the residuals by creating an extreme value probability plot of the residuals using the RELIABILITY procedure and the RESIDUAL data set. The following statements create the plot in Figure 37.19.

```
proc reliability data=residual;
   distribution ev;
   probplot sresid;
run;
```

The RELIABILITY Procedure

Model Information

Input Data Set	WORK.BEARING
Analysis Variable	life
Distribution	Weibull

Parameter Information

Parameter	Effect
Prm1	Intercept
Prm2	lload
Prm3	EV Scale

Algorithm converged.

Summary of Fit

Observations Used	39
Uncensored Values	39
Maximum Loglikelihood	-51.77737

Weibull Parameter Estimates

Parameter	Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits Lower	Upper
Intercept	0.8323	0.1410	0.5560	1.1086
lload	-13.8529	1.2333	-16.2703	-11.4356
EV Scale	0.8043	0.0999	0.6304	1.0260
Weibull Shape	1.2434	0.1545	0.9746	1.5862

Estimated Covariance Matrix
Weibull Parameters

	Prm1	Prm2	Prm3
Prm1	0.01987	-0.04374	-0.00492
Prm2	-0.04374	1.52113	0.01578
Prm3	-0.00492	0.01578	0.00999

Estimated Correlation Matrix
Weibull Parameters

	Prm1	Prm2	Prm3
Prm1	1.0000	-0.2516	-0.3491
Prm2	-0.2516	1.0000	0.1281
Prm3	-0.3491	0.1281	1.0000

Figure 37.17. Analysis Results for the Bearing Data

Obs	life	lload	Xbeta	Surv	Resid	SRESID	Aresid
1	1.67	-0.139262	2.7614742	0.9407681	-2.248651	-2.795921	-2.795921
2	2.2	-0.139262	2.7614742	0.9175782	-1.973017	-2.453205	-2.453205
3	2.51	-0.139262	2.7614742	0.9036277	-1.841191	-2.289296	-2.289296
4	3	-0.139262	2.7614742	0.8811799	-1.662862	-2.067565	-2.067565
5	3.9	-0.139262	2.7614742	0.8392186	-1.400498	-1.741347	-1.741347
6	4.7	-0.139262	2.7614742	0.8016738	-1.213912	-1.50935	-1.50935
7	7.53	-0.139262	2.7614742	0.6721971	-0.742579	-0.923306	-0.923306
8	14.7	-0.139262	2.7614742	0.4015113	-0.073627	-0.091546	-0.091546
9	27.76	-0.139262	2.7614742	0.1337746	0.562122	0.6989298	0.6989298
10	37.4	-0.139262	2.7614742	0.0542547	0.8601965	1.069549	1.069549
11	0.8	-0.01005	0.971511	0.7973909	-1.194655	-1.485407	-1.485407
12	1	-0.01005	0.971511	0.741702	-0.971511	-1.207955	-1.207955
13	1.37	-0.01005	0.971511	0.6427726	-0.6567	-0.816526	-0.816526
14	2.25	-0.01005	0.971511	0.4408692	-0.160581	-0.199663	-0.199663
15	2.95	-0.01005	0.971511	0.3175927	0.1102941	0.1371372	0.1371372
16	3.7	-0.01005	0.971511	0.2186832	0.3368218	0.4187966	0.4187966
17	6.07	-0.01005	0.971511	0.0600164	0.8318476	1.0343005	1.0343005
18	6.65	-0.01005	0.971511	0.0428027	0.9231058	1.147769	1.147769
19	7.05	-0.01005	0.971511	0.0337583	0.9815166	1.2203956	1.2203956
20	7.37	-0.01005	0.971511	0.0278531	1.0259067	1.2755892	1.2755892
21	0.18	0.0861777	-0.361531	0.8303684	-1.353268	-1.682623	-1.682623
22	0.2	0.0861777	-0.361531	0.809042	-1.247907	-1.55162	-1.55162
23	0.24	0.0861777	-0.361531	0.7665749	-1.065586	-1.324925	-1.324925
24	0.26	0.0861777	-0.361531	0.7455451	-0.985543	-1.225402	-1.225402
25	0.32	0.0861777	-0.361531	0.6837688	-0.777904	-0.967228	-0.967228
26	0.32	0.0861777	-0.361531	0.6837688	-0.777904	-0.967228	-0.967228
27	0.42	0.0861777	-0.361531	0.5868036	-0.50597	-0.629112	-0.629112
28	0.44	0.0861777	-0.361531	0.5684693	-0.45945	-0.57127	-0.57127
29	0.88	0.0861777	-0.361531	0.2625812	0.2336973	0.290574	0.290574
30	0.073	0.1655144	-1.460578	0.7887184	-1.156718	-1.438237	-1.438237
31	0.098	0.1655144	-1.460578	0.7101313	-0.86221	-1.072052	-1.072052
32	0.117	0.1655144	-1.460578	0.6526714	-0.685003	-0.851717	-0.851717
33	0.135	0.1655144	-1.460578	0.6006317	-0.541902	-0.673789	-0.673789
34	0.175	0.1655144	-1.460578	0.4946523	-0.282391	-0.351119	-0.351119
35	0.262	0.1655144	-1.460578	0.3126729	0.1211675	0.1506569	0.1506569
36	0.27	0.1655144	-1.460578	0.2991233	0.1512449	0.1880546	0.1880546
37	0.35	0.1655144	-1.460578	0.1889073	0.4107561	0.5107249	0.5107249
38	0.386	0.1655144	-1.460578	0.1522503	0.5086604	0.6324568	0.6324568
39	0.456	0.1655144	-1.460578	0.0987061	0.6753158	0.8396724	0.8396724

Figure 37.18. Listing of RESIDUAL

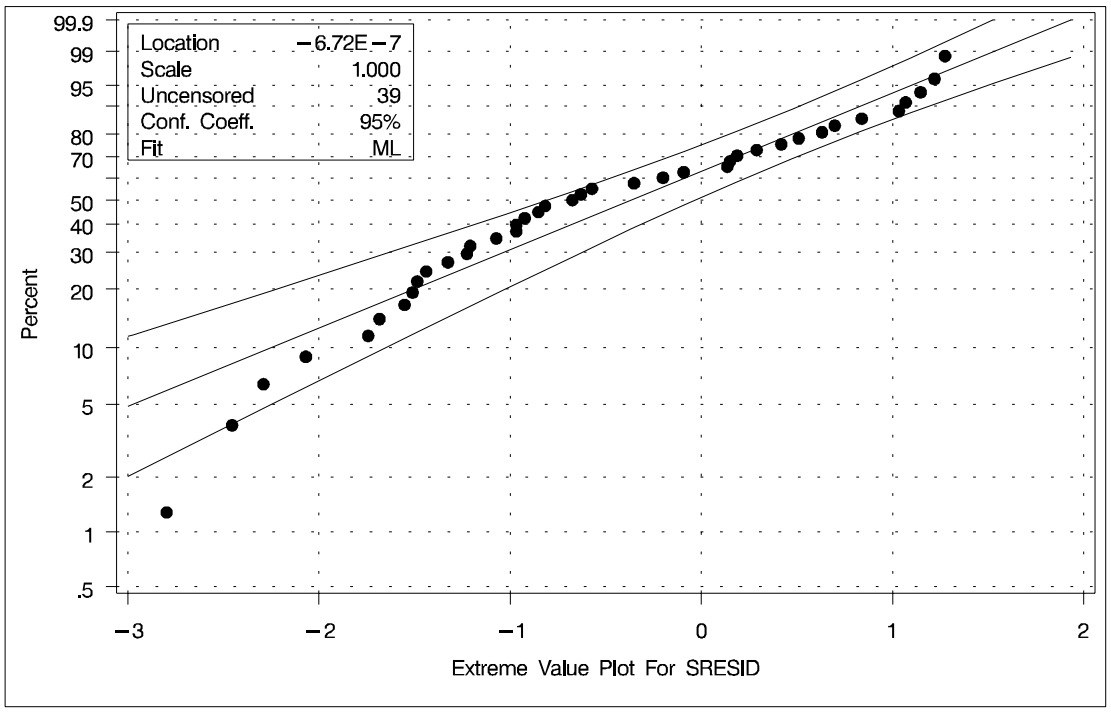


Figure 37.19. Extreme Value Probability Plot for the Standardized Residuals

Although the estimated location is near zero and the estimated scale is near one, the plot reveals systematic curvature, indicating that the Weibull regression model might be inadequate.

Regression Model with Non-Constant Scale

Nelson (1990, p. 272) and Meeker and Escobar (1998, p. 439) analyzed data from a strain-controlled fatigue test on 26 specimens of a type of superalloy. The following SAS statements create a SAS data set containing for each specimen the level of pseudo-stress (Pstress), the number of cycles (in thousands) (Kcycles) until failure or removal from the test, and a variable to indicate whether a specimen failed (F) or was right censored (C) (Status):

```
data alloy;
   input pstress kcycles status$ @@;
   cen = ( status = 'C' );
   datalines;
80.3  211.629  F    99.8   43.331  F
80.6  200.027  F   100.1   12.076  F
80.8   57.923  C   100.5   13.181  F
84.3  155.000  F   113.0   18.067  F
85.2   13.949  F   114.8   21.300  F
85.6  112.968  C   116.4   15.616  F
85.8  152.680  F   118.0   13.030  F
86.4  156.725  F   118.4    8.489  F
86.7  138.114  C   118.6   12.434  F
```

```
87.2   56.723  F   120.4    9.750  F
87.3  121.075  F   142.5   11.865  F
89.7  122.372  C   144.5    6.705  F
91.3  112.002  F   145.9    5.733  F
;

run;
```

The following statements fit a Weibull regression model with the number of cycles to failure as the response variable. The data set=RESIDS contains standardized residuals created with the ODS OUTPUT statement. The MODEL statement specifies a model quadratic in the log of pseudo-stress for the extreme value location parameter. The quadratic model in pseudo-stress PSTRESS is specified in the MODEL statement, and the RELATION=POW option specifies the log transformation be applied to Pstress in the MODEL statement and the LOGSCALE statement. The LOGSCALE statement specifies the log of the scale parameter as a linear function of the log of Pstress. The RPLOT statement specifies a plot of the data and the fitted regression model versus the variable Pstress. The FIT=REGRESSION option specifies plotting the regression model fitted with the preceding MODEL statement. The RELATION=POW option specifies using a log stress axis. The PLOTFIT option specifies plotting the 10th, 50th, and 90th percentiles of the regression model at each stress level. The SLOWER, SUPPER, and LUPPER options control limits on the stress and lifetime axes:

```
ods output ModObstats = Resids;
proc reliability data = alloy;
   distribution weibull;
   model kcycles*cen(1) = pstress pstress*pstress / Relation = Pow Obstats;
   logscale pstress;
   rplot kcycles*cen(1) = pstress /  fit=regression
                                     relation = pow
                                     plotfit  10 50 90
                                     slower=60 supper=160
                                     lupper=500;

   label pstress = "Pseudo-Stress";
   label kcycles = "Thousands of Cycles";
run;
```

Figure 37.20 displays the parameter estimates from the fitted regression model. Parameter estimates for both the model for the location parameter and the scale parameter models are shown. Standard errors and confidence limits for all parameter estimates are included.

The RELIABILITY Procedure

Weibull Parameter Estimates

Parameter	Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits Lower	Upper
Intercept	243.1681	58.0666	129.3596	356.9766
pstress	-96.5240	24.7075	-144.9498	-48.0983
pstress*pstress	9.6653	2.6247	4.5210	14.8095

Log-Scale Parameter Estimates

Parameter	Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits Lower	Upper
Intercept	4.4666	4.0724	-3.5152	12.4485
pstress	-1.1757	0.8731	-2.8870	0.5355

Figure 37.20. Parameter Estimates for Fitted Regression Model

Figure 37.21 displays the plot of the data and fitted regression model.

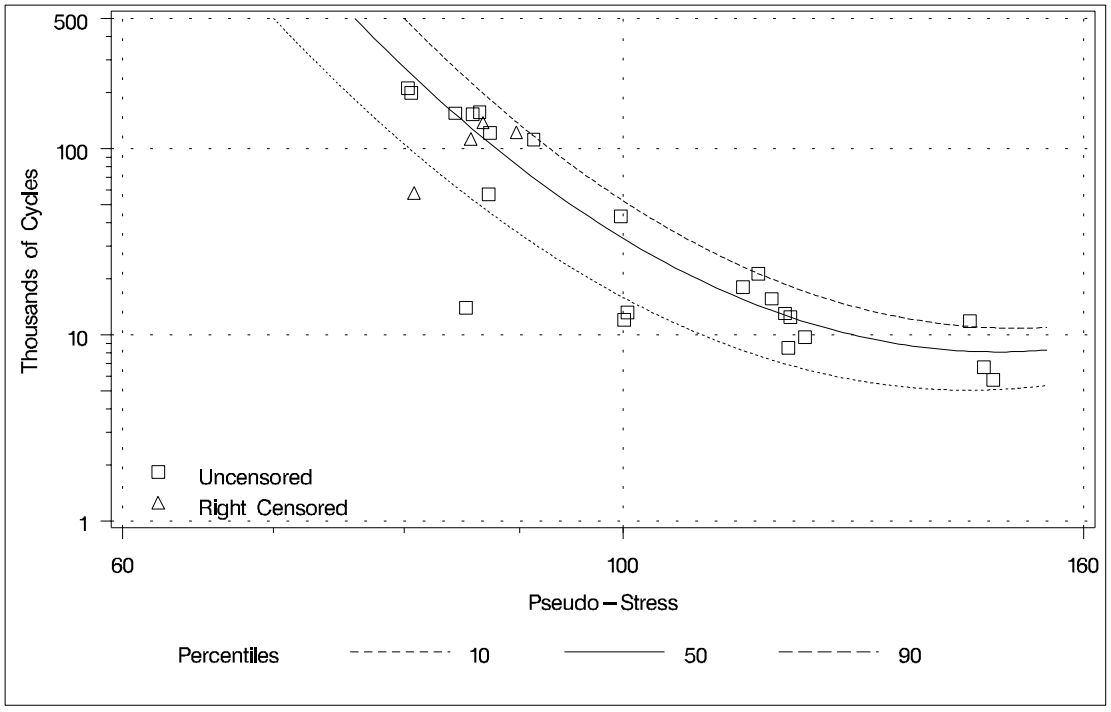


Figure 37.21. Superalloy Fatigue Data with Fitted Regression Model

The following SAS statements create an extreme values probability plot of standardized residuals from the regression model shown in Figure 37.22:

```
proc reliability data = Resids;
   distribution ev;
   pplot sresid*cen(1) / nofit
                         ;
run;
```

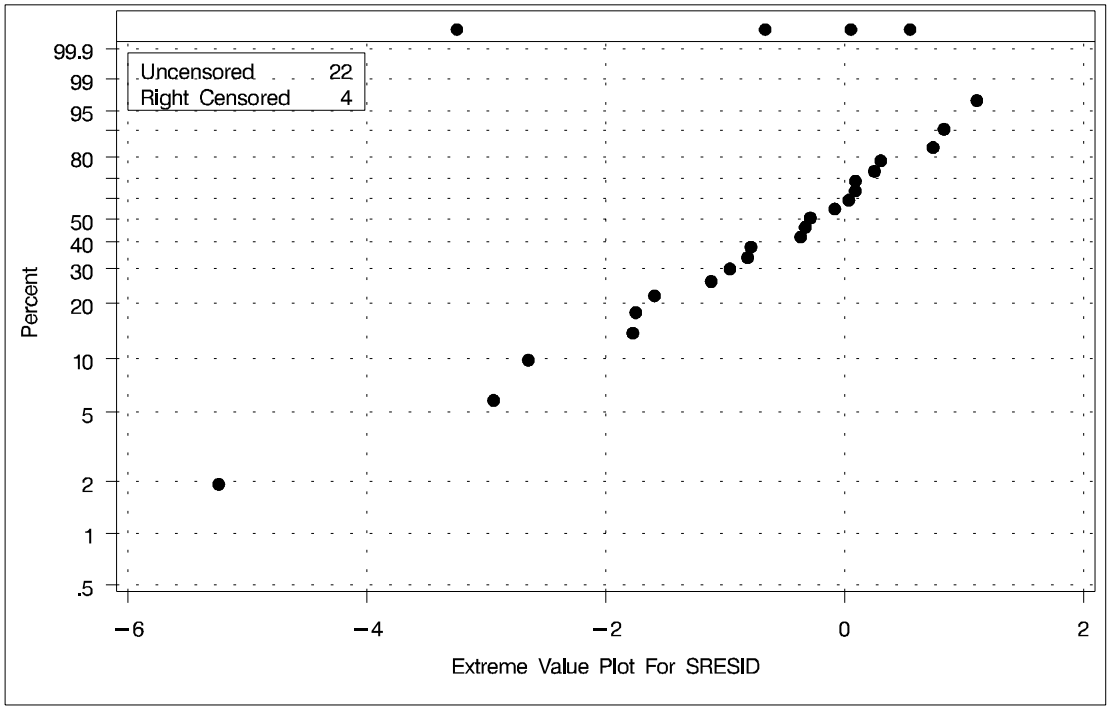


Figure 37.22. Residuals for Superalloy Fatigue Data Regression Model

Regression Model with Two Independent Variables

Meeker and Escobar (1998, p. 447) analyze data from an accelerated test on the lifetimes of glass capacitors as a function of operating voltage and temperature. The following SAS statements create a SAS data set containing the data. There are four lifetimes for each of eight combinations and four censored observations after the fourth failure for each combination:

```
data glass;
   input Temp Voltage @;
   do i = 1 to 4;
      cen = 0;
      input Hours @; output;
   end;
   do i = 1 to 4;
      cen = 1;
      output;
   end;
   datalines;
170 200  439  904  1092  1105
170 250  572  690   904  1090
170 300  315  315   439   628
170 350  258  258   347   588
180 200  959 1065  1065  1087
180 250  216  315   455   473
180 300  241  315   332   380
180 350  241  241   435   455
;
run;
```

The following statements analyze the capacitor data. The MODEL statement fits a regression model with Temp and Voltage as independent variables. Parameter estimates from the fitted regression model are shown in Figure 37.23. An interaction term between Temp and Voltage is included. The PPLOT statement creates a Weibull probability plot shown in Figure 37.24 with all temperature-voltage combinations overlaid on the same plot. The regression model fit is also plotted. The RPLOT statement creates the plot shown in Figure 37.25 of the data and Weibull distribution percentiles from the regression model as a function of voltage for values of temperature of 150, 170, and 180:

```
proc reliability data = glass;
   distribution Weibull;
   model Hours*cen(1) = temp voltage temp * voltage;
   pplot Hours*cen(1) = ( temp voltage ) / fit = model
                                           overlay
                                           noconf
                                           lupper = 2000
                                           lfit = ( 1 to 11 );


run;


proc reliability data = glass;
   distribution Weibull;
   model Hours*cen(1) = temp voltage temp * voltage;
   rplot Hours*cen(1) = voltage / fit = regression
                                  plotfit
                                  temp = 150, 170, 180;
run;
```

The RELIABILITY Procedure

Weibull Parameter Estimates

Parameter	Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits Lower	Upper
Intercept	9.4135	10.5402	-11.2449	30.0719
Temp	-0.0062	0.0598	-0.1235	0.1110
Voltage	0.0086	0.0374	-0.0648	0.0820
Temp*Voltage	-0.0001	0.0002	-0.0005	0.0003
EV Scale	0.3624	0.0553	0.2687	0.4887
Weibull Shape	2.7593	0.4210	2.0461	3.7209

Figure 37.23. Parameter Estimates for Fitted Regression Model

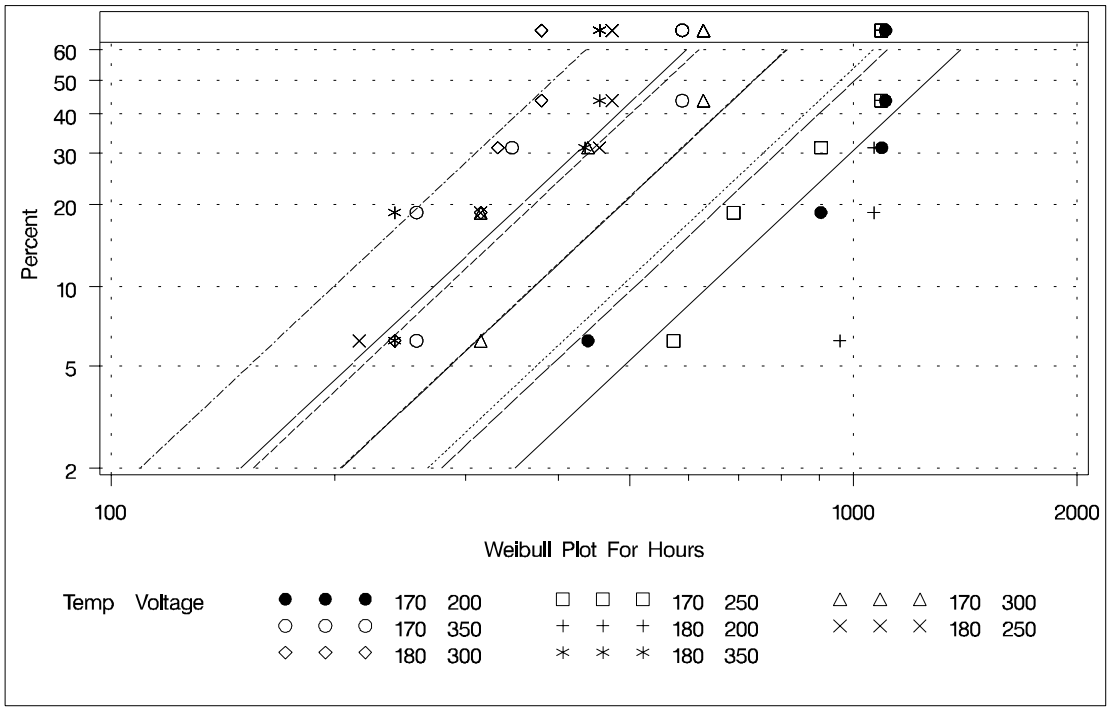


Figure 37.24. Probability Plot for Glass Capacitor Regression Model

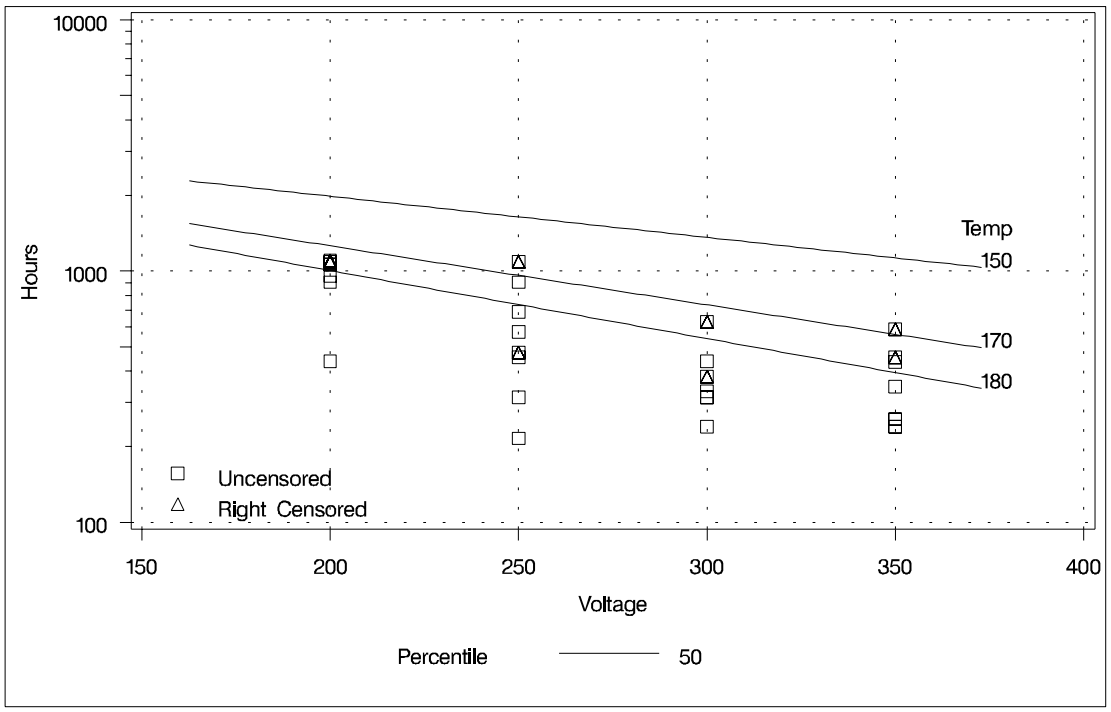


Figure 37.25. Plot of Data and Fitted Weibull Percentiles for Glass Capacitor Regression Model

Weibull Probability Plot for Two Combined Failure Modes

Doganaksoy, Hahn, and Meeker (2002) analyze failure data for the dielectric insulation of generator armature bars. A sample of 58 segments of bars were subjected to a high voltage stress test. Based on examination of the sample after the test, failures were attributed to one of two modes:

- Mode D (degradation failure): degradation of the organic material. Such failures usually occur later in life.
- Mode E (early failure): insulation defects due to a processing problem. These failures tend to occur early in life.

The following SAS statements create a SAS data set that contains the failure data. The variable Hours represents the number of hours until a failure, or the number of hours on test if the sample unit did not fail. The variable Mode represents the failure mode: D for degradation failure, E for early failures, or Cen if the unit did not fail, i.e., is right-censored. The computed variable Status is a numeric indicator for censored observations.

```
data Voltage;
   input Hours Mode$ @@;
   if Mode = 'Cen' then Status = 1;
   else Status = 0;
   datalines;
2   E   3   E   5   E   8   E   13  Cen 21  E
28  E   31  E   31  Cen 52  Cen 53  Cen 64  E
67  Cen 69  E   76  E   78  Cen 104 E   113 Cen
119 E   135 Cen 144 E   157 Cen 160 E   168 D
179 Cen 191 D   203 D   211 D   221 E   226 D
236 E   241 Cen 257 Cen 261 D   264 D   278 D
282 E   284 D   286 D   298 D   303 E   314 D
317 D   318 D   320 D   327 D   328 D   328 D
348 D   348 Cen 350 D   360 D   369 D   377 D
387 D   392 D   412 D   446 D
;
run;
```

The following statements fit a Weibull distribution to the individual failure modes (D and E), and compute the failure distribution with both modes acting.

```
proc reliability data=Voltage;
   distribution Weibull;
   pplot Hours*Status(1) / vref(intersect) = 10
                           vreflabel = ('10% Percentile')
                           survtime = 100 200 300 400 500 1000
                           lupper = 500;
   fmode combine = Mode( D E );
run;
```

Figure 37.26 contains estimates of the combined failure mode survival function at the times specified with the SURVTIME= option in the PPLOT statement.

The RELIABILITY Procedure

Combined Failure Modes
Weibull Distribution Function Estimates
With 95% Asymptotic Normal Confidence Limits

X	Pr(<X)	Lower	Upper	Pr(>X)	Lower	Upper
100.00	0.1898	0.1172	0.2926	0.8102	0.7074	0.8828
200.00	0.3115	0.2139	0.4292	0.6885	0.5708	0.7861
300.00	0.5866	0.4621	0.7010	0.4134	0.2990	0.5379
400.00	0.9405	0.8476	0.9782	0.0595	0.0218	0.1524
500.00	0.9998	0.9711	1.0000	0.0002	0.0000	0.0289
1000.00	1.0000	0.0000	1.0000	0.0000	0.0000	1.0000

Figure 37.26. Survival Function Estimates for Combined Failure Modes

Figure 37.27 shows Weibull parameter estimates for the two individual failure modes.

Individual Failure Mode
Weibull Parameter Estimates

Parameter	Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits Lower	Upper	Mode
EV Location	5.8415	0.0350	5.7730	5.9100	D
EV Scale	0.1785	0.0254	0.1350	0.2360	D
Weibull Scale	344.2966	12.0394	321.4903	368.7208	D
Weibull Shape	5.6020	0.7985	4.2365	7.4076	D
EV Location	7.0649	0.5109	6.0637	8.0662	E
EV Scale	1.5739	0.3415	1.0287	2.4080	E
Weibull Scale	1170.1832	597.7903	429.9480	3184.8703	E
Weibull Shape	0.6354	0.1379	0.4153	0.9721	E

Figure 37.27. Parameter Estimates for Individual Failure Modes

Figure 37.28 is a Weibull probability plot of the failure probability distribution with the two failure modes combined, along with approximate pointwise 95% confidence limits. A reference line at the 10% point on the vertical axis intersecting the distribution curve shows the 10% percentile of lifetimes when both modes act to be about 34 hours.

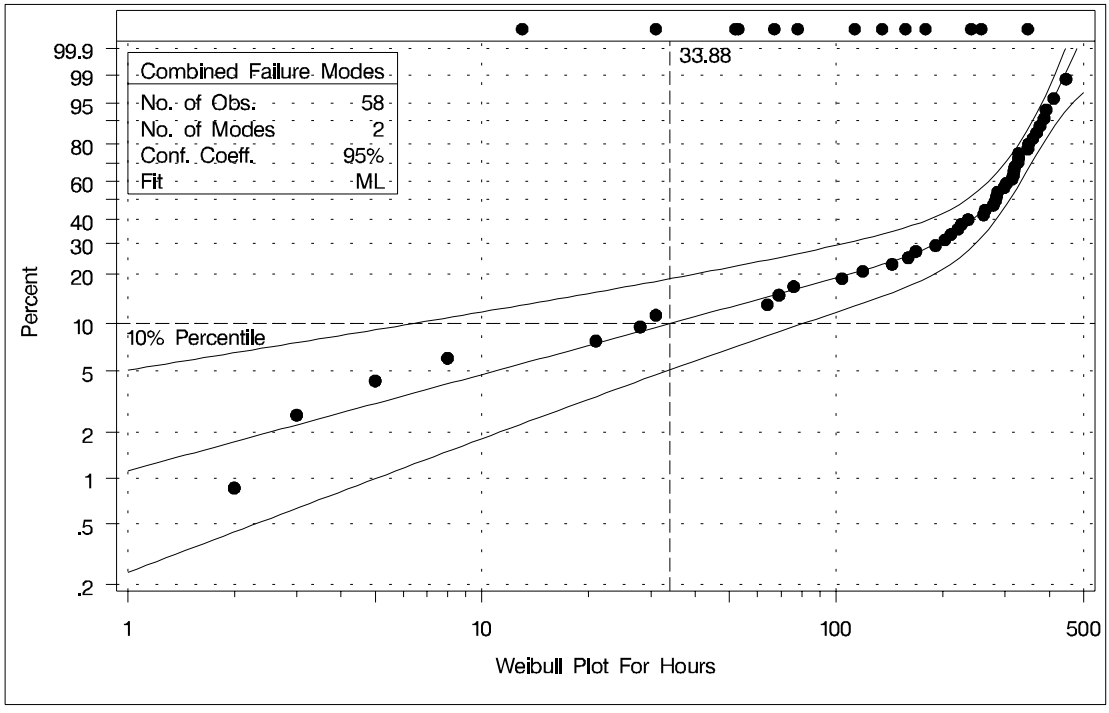


Figure 37.28. Weibull Plot for Failure Modes D and E

The following SAS statements create the Weibull probability plot in Figure 37.29. The PLOTMODES option in the FMODE statement cause the Weibull fits for the individual failure modes to be included on the probability plot. The combined failure mode curve is almost the same as the fit for mode E for lifetimes of less than 100 hours, and slightly above the fit for mode D for higher lifetimes.

```
proc reliability data=Voltage;
   distribution Weibull;
   pplot Hours*Status(1) / vref(intersect) = 10
                           vreflabel = ('10% Percentile')
                           survtime = 100 200 300 400 500 1000
                           noconf
                           lupper = 500;
   fmode combine = Mode( D E ) / plotmodes;
run;
```

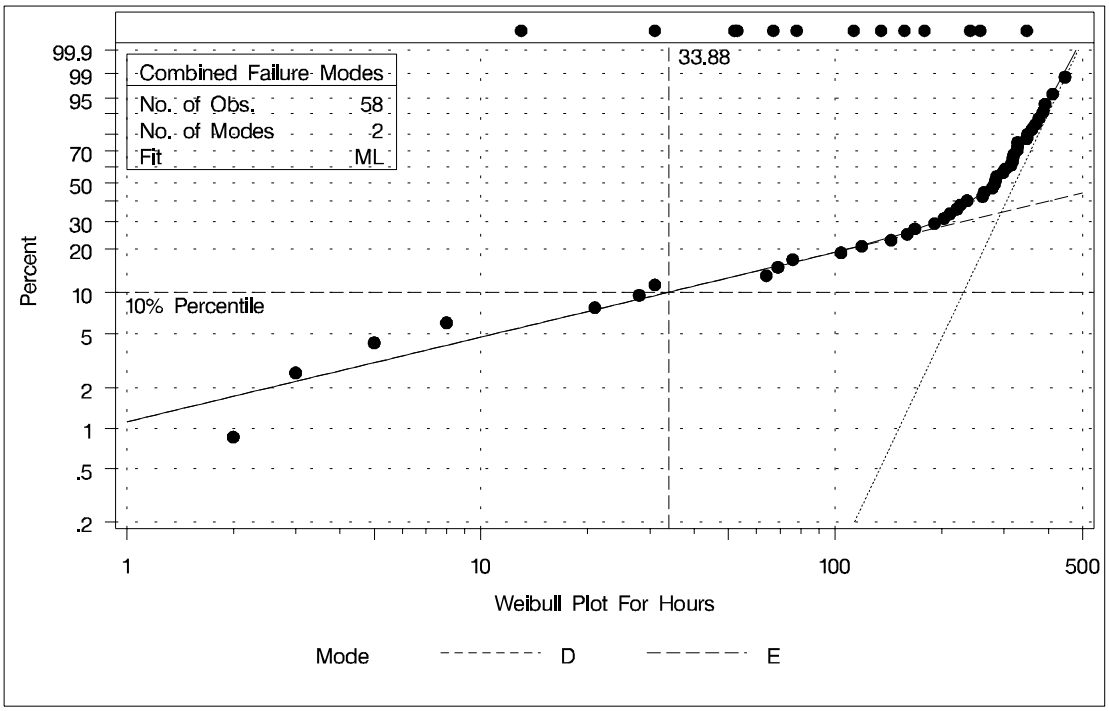


Figure 37.29. Weibull Plot for Failure Modes D and E With Individual Modes

Analysis of Recurrence Data on Repairs

This example illustrates analysis of recurrence data from repairable systems. Repair data analysis differs from life data analysis, where units fail only once. As a repairable system ages, it accumulates repairs and costs of repairs. The RELIABILITY procedure provides a nonparametric estimate and plot of the *mean cumulative function* (MCF) for the number or cost of repairs for a population of repairable systems.

The nonparametric estimate of the MCF, the variance of the MCF estimate, and confidence limits for the MCF estimate are based on the work of Nelson (1995). The MCF, also written as $M(t)$, is defined by Nelson (1995) to be the *population mean* of the distribution of the cumulative number or cost of repairs at age t. The method does not assume any underlying structure for the repair process.

The SAS statements that follow create the listing of the SAS data set VALVE shown in Figure 37.30, which contains repair histories of 41 diesel engines in a fleet (Nelson 1995). The valve seats in these engines wear out and must be replaced. The variable Id is a unique identifier for individual engines. The variable Days provides the engine age in days. The value of the variable Value is 1 if the age is a valve seat replacement age or -1 if the age is the end of history, or censoring age, for the engine.

```
data valve;
   input id days value @@;
   datalines;
251 761 -1      252 759 -1      327  98  1      327 667 -1
328 326  1      328 653  1      328 653  1      328 667 -1
329 665 -1      330  84  1      330 667 -1      331  87  1
331 663 -1      389 646  1      389 653 -1      390  92  1
390 653 -1      391 651 -1      392 258  1      392 328  1
392 377  1      392 621  1      392 650 -1      393  61  1
393 539  1      393 648 -1      394 254  1      394 276  1
394 298  1      394 640  1      394 644 -1      395  76  1
395 538  1      395 642 -1      396 635  1      396 641 -1
397 349  1      397 404  1      397 561  1      397 649 -1
398 631 -1      399 596 -1      400 120  1      400 479  1
400 614 -1      401 323  1      401 449  1      401 582 -1
402 139  1      402 139  1      402 589 -1      403 593 -1
404 573  1      404 589 -1      405 165  1      405 408  1
405 604  1      405 606 -1      406 249  1      406 594 -1
407 344  1      407 497  1      407 613 -1      408 265  1
408 586  1      408 595 -1      409 166  1      409 206  1
409 348  1      409 389 -1      410 601 -1      411 410  1
411 581  1      411 601 -1      412 611 -1      413 608 -1
414 587 -1      415 367  1      415 603 -1      416 202  1
416 563  1      416 570  1      416 585 -1      417 587 -1
418 578 -1      419 578 -1      420 586 -1      421 585 -1
422 582 -1
;
run;
```

Obs	id	days	value
1	251	761	-1
2	252	759	-1
3	327	98	1
4	327	667	-1
5	328	326	1
6	328	653	1
7	328	653	1
8	328	667	-1
9	329	665	-1
10	330	84	1
11	330	667	-1

Figure 37.30. Partial Listing of the Valve Seat Data

The following statements produce the graphical display in Figure 37.31.

```
proc reliability;
  unitid id;
  mcfplot days*value(-1) / plotsymbol = X
                           nocenprint;
run;
```

The UNITID statement specifies that the variable Id uniquely identifies each system. The MCFPLOT statement requests a plot of the MCF estimates as a function of the age variable Days, and it specifies -1 as the value of the variable Value, which identifies the end of history for each engine (system). The option NOCENPRINT specifies that only failure times, and not censoring times, be printed in the tabular output.

In Figure 37.31, the MCF estimates and confidence limits are plotted versus system age in days. The end-of-history ages are plotted in an area at the top of the plot. Except for the last few points, the plot is essentially a straight line, suggesting a constant replacement rate. Consequently, the prediction of future replacements of valve seats can be based on a fitted line in this case.

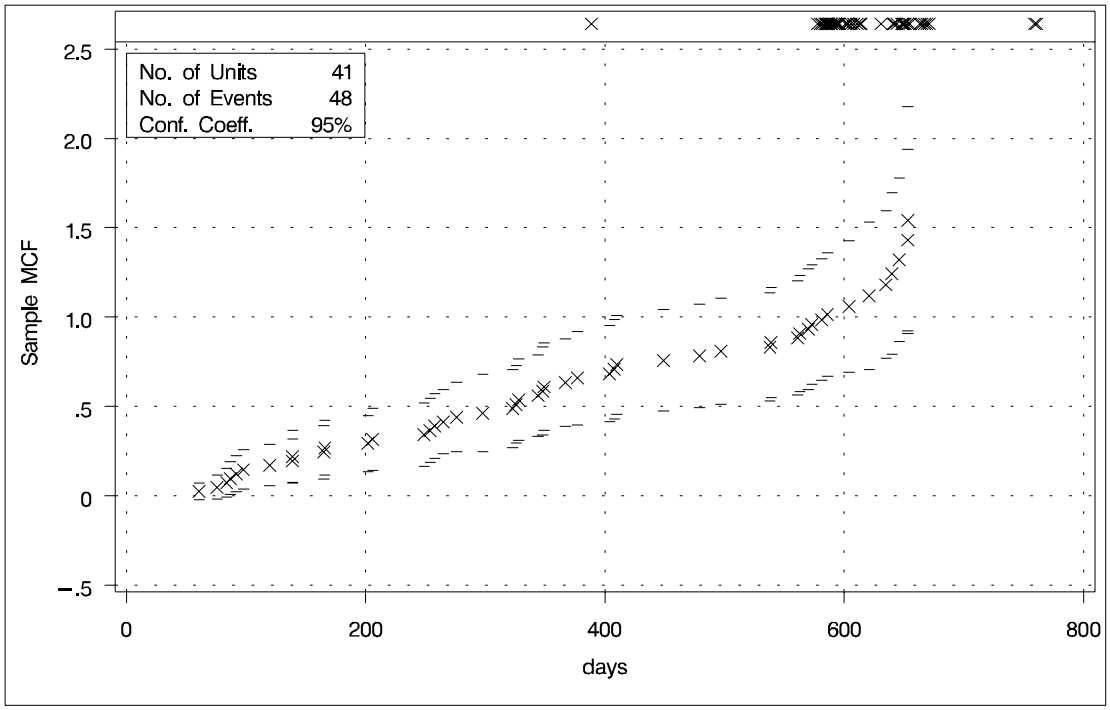


Figure 37.31. Mean Cumulative Function for the Number of Repairs

A partial listing of the tabular output is shown in Figure 37.32 and Figure 37.33. It contains a summary of the repair data, estimates of the MCF, the Nelson (1995) standard errors, and confidence intervals for the MCF.

Recurrence Data Summary

Input Data Set	WORK.VALVE
Observations Used	89
Number of Units	41
Number of Events	48

Figure 37.32. Partial Listing of the Output for the Valve Seat Data

Recurrence Data Analysis

Age	Sample MCF	Standard Error	95% Confidence Limits Lower	Upper	Unit ID
61.00	0.024	0.024	-0.023	0.072	393
76.00	0.049	0.034	-0.018	0.116	395
84.00	0.073	0.041	-0.008	0.154	330
87.00	0.098	0.047	0.006	0.190	331
92.00	0.122	0.052	0.021	0.223	390
98.00	0.146	0.056	0.037	0.256	327
120.00	0.171	0.059	0.054	0.287	400
139.00	0.195	0.063	0.072	0.318	402
139.00	0.220	0.074	0.074	0.365	402
165.00	0.244	0.076	0.094	0.394	405
166.00	0.268	0.078	0.115	0.422	409
202.00	0.293	0.080	0.136	0.449	416
206.00	0.317	0.089	0.143	0.491	409
249.00	0.341	0.090	0.165	0.517	406
254.00	0.366	0.091	0.188	0.544	394
258.00	0.390	0.092	0.211	0.570	392
265.00	0.415	0.092	0.234	0.595	408
276.00	0.439	0.099	0.245	0.633	394
298.00	0.463	0.111	0.246	0.681	394
323.00	0.488	0.111	0.270	0.706	401
326.00	0.512	0.111	0.294	0.730	328
328.00	0.537	0.116	0.309	0.765	392
344.00	0.561	0.116	0.333	0.788	407
348.00	0.585	0.126	0.339	0.832	409
349.00	0.610	0.125	0.364	0.855	397
367.00	0.634	0.125	0.390	0.879	415
377.00	0.659	0.133	0.397	0.920	392
404.00	0.684	0.138	0.414	0.953	397
408.00	0.709	0.142	0.431	0.986	405
410.00	0.734	0.141	0.457	1.010	411
449.00	0.759	0.144	0.475	1.042	401
479.00	0.784	0.148	0.494	1.073	400
497.00	0.809	0.151	0.512	1.105	407
538.00	0.834	0.154	0.531	1.136	395
539.00	0.859	0.157	0.551	1.166	393
561.00	0.884	0.164	0.563	1.205	397
563.00	0.909	0.166	0.583	1.234	416
570.00	0.934	0.172	0.596	1.272	416
573.00	0.959	0.171	0.623	1.294	404
581.00	0.985	0.173	0.645	1.325	411
586.00	1.014	0.176	0.669	1.359	408
604.00	1.060	0.188	0.692	1.427	405
621.00	1.119	0.211	0.705	1.532	392
635.00	1.181	0.210	0.768	1.594	396
640.00	1.244	0.231	0.791	1.696	394
646.00	1.320	0.233	0.864	1.777	389
653.00	1.432	0.259	0.923	1.940	328
653.00	1.543	0.324	0.908	2.177	328

Figure 37.33. Partial Listing of the Output for the Valve Seat Data

Parametric modeling of the repair process requires more assumptions than nonparametric modeling, and considerable work has been done in this area. Ascher and Feingold (1984) describe parametric models for repair processes. For example, repairs are sometimes modeled as a nonhomogeneous Poisson process. The current release of the RELIABILITY procedure does not include this type of parametric mod-

eling, although it is planned for future releases. The MCF plot might be a first step in modeling a repair process, but, in many cases, it provides the required answers without further analysis. An estimate of the MCF for a sample of systems aids engineers in determining the repair rate at any age and the increase or decrease of repair rate with population age. The estimate is also useful for predicting the number of future repairs.

Comparison of Two Samples of Repair Data

Nelson (1995) and Doganaksoy and Nelson (1991) show how the difference of MCFs from two samples can be used to compare the populations from which they are drawn. The RELIABILITY procedure provides Doganaksoy and Nelson's confidence intervals for the pointwise difference of the two MCFs, which can be used to assess whether the difference is statistically significant.

Doganaksoy and Nelson (1991) give an example of two samples of locomotives with braking grids from two different production batches. Figure 37.34 contains a listing of the data. The variable ID is a unique identifier for individual locomotives. The variable Days provides the locomotive age in days. The variable Value is 1 if the age corresponds to a valve seat replacement or -1 if the age corresponds to the locomotive's latest age (the current end of its history). The variable Sample is a group variable that identifies the grid production batch.

```
data grids;
   if _N_ < 40 then sample = 'Sample1';
   else sample = 'Sample2';
   input ID$ days value @@;
   cards;
S1-01 462  1    S1-01 730 -1    S1-02 364  1    S1-02 391  1
S1-02 548  1    S1-02 724 -1    S1-03 302  1    S1-03 444  1
S1-03 500  1    S1-03 730 -1    S1-04 250  1    S1-04 730 -1
S1-05 500  1    S1-05 724 -1    S1-06  88  1    S1-06 724 -1
S1-07 272  1    S1-07 421  1    S1-07 552  1    S1-07 625  1
S1-07 719 -1    S1-08 481  1    S1-08 710 -1    S1-09 431  1
S1-09 710 -1    S1-10 367  1    S1-10 710 -1    S1-11 635  1
S1-11 650  1    S1-11 708 -1    S1-12 402  1    S1-12 700 -1
S1-13  33  1    S1-13 687 -1    S1-14 287  1    S1-14 687 -1
S1-15 317  1    S1-15 498  1    S1-15 657 -1    S2-01 203  1
S2-01 211  1    S2-01 277  1    S2-01 373  1    S2-01 511 -1
S2-02 293  1    S2-02 503 -1    S2-03 173  1    S2-03 470 -1
S2-04 242  1    S2-04 464 -1    S2-05  39  1    S2-05 464 -1
S2-06  91  1    S2-06 462 -1    S2-07 119  1    S2-07 148  1
S2-07 306  1    S2-07 461 -1    S2-08 382  1    S2-08 460 -1
S2-09 250  1    S2-09 434 -1    S2-10 192  1    S2-10 448 -1
S2-11 369  1    S2-11 448 -1    S2-12  22  1    S2-12 447 -1
S2-13  54  1    S2-13 441 -1    S2-14 194  1    S2-14 432 -1
S2-15  61  1    S2-15 419 -1    S2-16  19  1    S2-16 185  1
S2-16 419 -1    S2-17 187  1    S2-17 416 -1    S2-18  93  1
S2-18 205  1    S2-18 264  1    S2-18 415 -1
;
```

Obs	sample	ID	days	value
1	Sample1	S1-01	462	1
2	Sample1	S1-01	730	-1
3	Sample1	S1-02	364	1
4	Sample1	S1-02	391	1
5	Sample1	S1-02	548	1
6	Sample1	S1-02	724	-1
7	Sample1	S1-03	302	1
8	Sample1	S1-03	444	1
9	Sample1	S1-03	500	1
10	Sample1	S1-03	730	-1
11	Sample1	S1-04	250	1
12	Sample1	S1-04	730	-1
13	Sample1	S1-05	500	1
14	Sample1	S1-05	724	-1
15	Sample1	S1-06	88	1
16	Sample1	S1-06	724	-1
17	Sample1	S1-07	272	1
18	Sample1	S1-07	421	1
19	Sample1	S1-07	552	1
20	Sample1	S1-07	625	1
21	Sample1	S1-07	719	-1
22	Sample1	S1-08	481	1
23	Sample1	S1-08	710	-1
24	Sample1	S1-09	431	1
25	Sample1	S1-09	710	-1
65	Sample2	S2-11	369	1
66	Sample2	S2-11	448	-1
67	Sample2	S2-12	22	1
68	Sample2	S2-12	447	-1
69	Sample2	S2-13	54	1
70	Sample2	S2-13	441	-1
71	Sample2	S2-14	194	1
72	Sample2	S2-14	432	-1
73	Sample2	S2-15	61	1
74	Sample2	S2-15	419	-1
75	Sample2	S2-16	19	1
76	Sample2	S2-16	185	1
77	Sample2	S2-16	419	-1
78	Sample2	S2-17	187	1
79	Sample2	S2-17	416	-1
80	Sample2	S2-18	93	1
81	Sample2	S2-18	205	1
82	Sample2	S2-18	264	1
83	Sample2	S2-18	415	-1

Figure 37.34. Partial Listing of the Braking Grids Data

The following statements request the Nelson (1995) nonparametric estimate and confidence limits for the difference of the MCF functions shown in Figure 37.35 for the braking grids.

```
proc reliability data=grids;
   unitid ID;
   mcfplot days*value(-1) = sample / mcfdiff
                                     plotsymbol = X
                                   ;
run;
```

The MCFPLOT statement requests a plot of each MCF estimate as a function of age (provided by Days), and it specifies that the end of history for each system is

identified by Value equal to -1. The variable Sample identifies the two samples of braking grids. The option MCFDIFF requests that the difference between the MCFs of the two groups given in the variable Sample be computed and plotted. Confidence limits for the MCF difference are also computed and plotted. The UNITID statement specifies that the variable Id uniquely identifies each system.

Figure 37.35 shows the plot of the MCF difference function and pointwise 95% confidence intervals. Since the pointwise confidence limits do not include zero for some system ages, the difference between the two populations is statistically significant.

A listing of the tabular output is shown in Figure 37.36. It contains a summary of the repair data for the two samples, estimates, standard errors, and confidence intervals for the MCF difference.

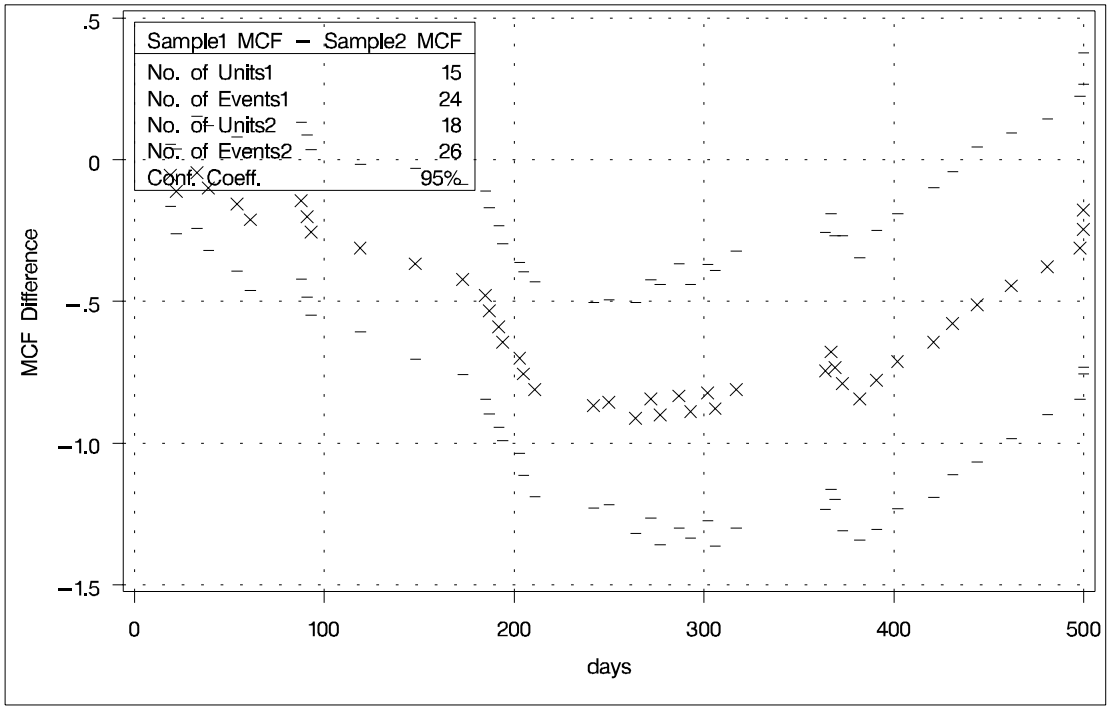


Figure 37.35. Mean Cumulative Function Difference

MCF Difference Data Summary

Input Data Set	WORK.GRIDS
Group 1	Sample1
Observations Used	39
Number of Units	15
Number of Events	24
Group 2	Sample2
Observations Used	44
Number of Units	18
Number of Events	26

Sample MCF Differences

Age	MCF Difference	Standard Error	95% Confidence Limits Lower	Upper	Unit ID
19.00	-0.056	0.056	-0.164	0.053	S2-16
22.00	-0.111	0.076	-0.261	0.038	S2-12
33.00	-0.044	0.101	-0.243	0.154	S1-13
39.00	-0.100	0.112	-0.320	0.120	S2-05
54.00	-0.156	0.121	-0.392	0.081	S2-13
61.00	-0.211	0.127	-0.461	0.039	S2-15
88.00	-0.144	0.142	-0.422	0.133	S1-06
91.00	-0.200	0.146	-0.486	0.086	S2-06
93.00	-0.256	0.149	-0.548	0.037	S2-18
119.00	-0.311	0.151	-0.607	-0.015	S2-07
148.00	-0.367	0.172	-0.703	-0.030	S2-07
173.00	-0.422	0.171	-0.758	-0.087	S2-03
185.00	-0.478	0.188	-0.846	-0.110	S2-16
187.00	-0.533	0.185	-0.897	-0.170	S2-17
192.00	-0.589	0.182	-0.946	-0.232	S2-10
194.00	-0.644	0.177	-0.992	-0.297	S2-14
203.00	-0.700	0.172	-1.037	-0.363	S2-01
205.00	-0.756	0.183	-1.115	-0.396	S2-18
211.00	-0.811	0.194	-1.191	-0.432	S2-01
242.00	-0.867	0.185	-1.230	-0.503	S2-04
250.00	-0.856	0.185	-1.218	-0.494	S1-04,S2-09
264.00	-0.911	0.208	-1.319	-0.503	S2-18
272.00	-0.844	0.214	-1.264	-0.424	S1-07
277.00	-0.900	0.234	-1.359	-0.441	S2-01
287.00	-0.833	0.238	-1.300	-0.367	S1-14
293.00	-0.889	0.228	-1.337	-0.441	S2-02
302.00	-0.822	0.231	-1.275	-0.369	S1-03
306.00	-0.878	0.248	-1.364	-0.391	S2-07
317.00	-0.811	0.250	-1.300	-0.322	S1-15
364.00	-0.744	0.250	-1.233	-0.255	S1-02
367.00	-0.678	0.248	-1.164	-0.191	S1-10
369.00	-0.733	0.237	-1.199	-0.268	S2-11
373.00	-0.789	0.265	-1.309	-0.269	S2-01
382.00	-0.844	0.254	-1.342	-0.347	S2-08
391.00	-0.778	0.269	-1.306	-0.250	S1-02
402.00	-0.711	0.266	-1.232	-0.190	S1-12
421.00	-0.644	0.279	-1.191	-0.098	S1-07
431.00	-0.578	0.273	-1.113	-0.043	S1-09
444.00	-0.511	0.283	-1.066	0.044	S1-03
462.00	-0.444	0.275	-0.984	0.095	S1-01
481.00	-0.378	0.266	-0.899	0.143	S1-08
498.00	-0.311	0.273	-0.846	0.224	S1-15
500.00	-0.244	0.261	-0.756	0.267	S1-05
500.00	-0.178	0.283	-0.733	0.377	S1-03

Figure 37.36. Listing of the Output for the Braking Grids Data

Analysis of Interval Age Recurrence Data

You can analyze recurrence data when the recurrence ages are grouped into intervals, instead of being exact ages. Figure 37.37 shows a listing of a SAS data set containing field data on replacements of defrost controls in 22,914 refrigerators, whose ages are grouped by months in service. Nelson (2002, Problem 5.2, Chapter 5) presents these data. Grouping the control data on the 22,914 refrigerators into age intervals enables you to represent the data by 29 data records, instead of requiring a single data record for each refrigerator, as required for exact recurrence data.

The variables Lower and Upper are the lower and upper monthly interval endpoints, Recurrences is the number of defrost control replacements in each month, and Censored is the number of refrigerator histories censored in each month, that is, the number with current age in the monthly interval. Data are entered as shown on Figure 37.37.

Obs	Lower	Upper	Recurrences	Censored
1	0	1	83	0
2	1	2	35	0
3	2	3	23	0
4	3	4	15	0
5	4	5	22	0
6	5	6	16	3
7	6	7	13	36
8	7	8	12	24
9	8	9	15	29
10	9	10	15	37
11	10	11	24	40
12	11	12	12	20041
13	12	13	7	14
14	13	14	11	17
15	14	15	15	13
16	15	16	6	28
17	16	17	8	22
18	17	18	9	27
19	18	19	9	64
20	19	20	5	94
21	20	21	6	119
22	21	22	6	118
23	22	23	6	138
24	23	24	5	1188
25	24	25	7	17
26	25	26	5	28
27	26	27	5	99
28	27	28	6	128
29	28	29	3	590

Figure 37.37. Listing of the Defrost Controls Data

The following SAS statements create the plot of the sample MCF of defrost control replacement shown in Figure 37.38 and the tabular listing in Figure 37.39. Pointwise confidence limits are included on the plot and in the tabular listing. These limits are approximate, and are usually shorter than the correct limits, which have not been developed for interval data.

```
proc reliability data=defrost;
   mcfplot ( interval    = Lower Upper
             recurrences = Recurrences
             censor      = Censored ) / plotsymbol = X
                                        vaxis = 0 to .12 by .04
                                        interpolate = join;
run;
```

Here, INTERVAL = LOWER UPPER specifies the input data set variables Lower and Upper as the age interval endpoints. The variable Recurrences identifies the number of recurrences (defrost control replacements) in each time interval, and Censored identifies the number of units censored in each interval (number in an age interval or removed from the sample in an age interval).

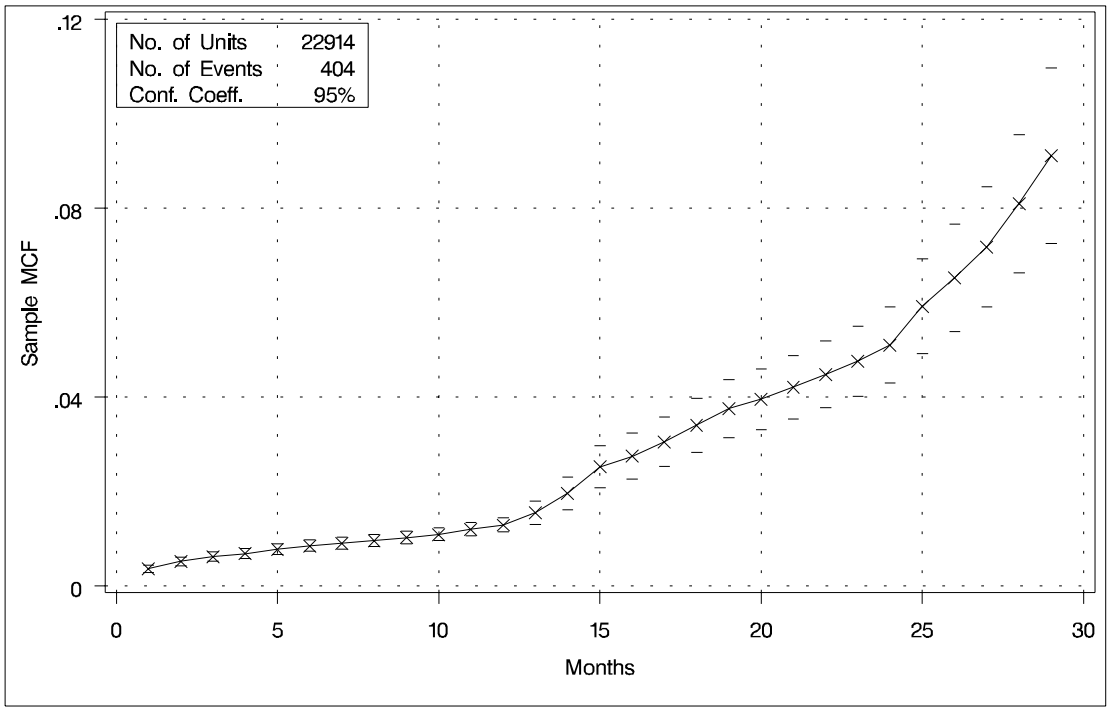


Figure 37.38. MCF Plot for the Defrost Controls

The last interval is always marked with a footnote indicating that estimates for the last interval may be biased since censoring ages often are not uniformly spread over that interval.

Recurrence Data Summary

Input Data Set	WORK.DEFROST
Observations Used	29
Number of Units	22914
Number of Events	404

Recurrence Data Analysis

	Endpoints		Sample	Standard	Naive 95% Confidence Limits	
	Lower	Upper	MCF	Error	Lower	Upper
	0.00	1.00	0.004	0.000	0.003	0.004
	1.00	2.00	0.005	0.000	0.004	0.006
	2.00	3.00	0.006	0.001	0.005	0.007
	3.00	4.00	0.007	0.001	0.006	0.008
	4.00	5.00	0.008	0.001	0.007	0.009
	5.00	6.00	0.008	0.001	0.007	0.010
	6.00	7.00	0.009	0.001	0.008	0.010
	7.00	8.00	0.010	0.001	0.008	0.011
	8.00	9.00	0.010	0.001	0.009	0.012
	9.00	10.00	0.011	0.001	0.010	0.012
	10.00	11.00	0.012	0.001	0.011	0.013
	11.00	12.00	0.013	0.001	0.011	0.014
	12.00	13.00	0.015	0.001	0.013	0.018
	13.00	14.00	0.020	0.002	0.016	0.023
	14.00	15.00	0.025	0.002	0.021	0.030
	15.00	16.00	0.027	0.002	0.023	0.032
	16.00	17.00	0.031	0.003	0.025	0.036
	17.00	18.00	0.034	0.003	0.028	0.040
	18.00	19.00	0.038	0.003	0.031	0.044
	19.00	20.00	0.040	0.003	0.033	0.046
	20.00	21.00	0.042	0.003	0.035	0.049
	21.00	22.00	0.045	0.004	0.038	0.052
	22.00	23.00	0.048	0.004	0.040	0.055
	23.00	24.00	0.051	0.004	0.043	0.059
	24.00	25.00	0.059	0.005	0.049	0.069
	25.00	26.00	0.065	0.006	0.054	0.077
	26.00	27.00	0.072	0.006	0.059	0.084
	27.00	28.00	0.081	0.007	0.066	0.096
*	28.00	29.00	0.091	0.010	0.072	0.110

* The estimate and limits for this interval may not be appropriate.

Figure 37.39. Listing of the Output for the Defrost Controls Data

Analysis of Binomial Data

This example illustrates the analysis of binomial proportions using capacitor failure data from nine circuit boards given by Nelson (1982, p. 451). The following statements create and list a SAS data set named BINEX containing the data.

```
data binex;
   input board sample fail;
   cards;
   1 84 2
   2 72 3
   3 72 5
   4 119 19
   5 538 21
   6 51 2
   7 517 9
   8 462 18
   9 143 2
   ;
   run;
```

Figure 37.40 displays a listing of the data. The variable Board identifies the circuit board, the variable Sample provides the number of capacitors on the boards, and the variable Fail provides the number of capacitors failing on the boards.

Obs	board	sample	fail
1	1	84	2
2	2	72	3
3	3	72	5
4	4	119	19
5	5	538	21
6	6	51	2
7	7	517	9
8	8	462	18
9	9	143	2

Figure 37.40. Listing of the Capacitor Data

The following statements analyze the proportion of capacitors failing.

```
proc reliability;
   distribution binomial;
   analyze fail(sample) = board / predict(1000)
                                  tolerance(.05);
run;
```

The DISTRIBUTION statement specifies the binomial distribution. The analysis requested with the ANALYZE statement consists of tabular output only. Graphical output is not available for the binomial distribution. The variable Fail provides the number of capacitors failing on each board, the variable Sample provides the sample

size (number of capacitors) for each board, and the variable Board identifies the individual boards. The statement option PREDICT(1000) requests the predicted number of capacitors failing and prediction limits in a future sample of size 1000. The option TOLERANCE(.05) requests the sample size required to estimate the binomial proportion to within 0.05. Figure 37.41 displays the results of the analysis.

The "Pooled Data Analysis" table displays the estimated binomial probability and exact binomial confidence limits when data from all boards are pooled. The chi-squared value and p-value for a test of equality of the binomial probabilities for all of the boards are also shown. In this case, the p-value is less than 0.05, so you reject the test of equality at the 0.05 level.

The "Predicted Values and Limits" table provides the predicted failure count and prediction limits for the number of capacitors that would fail in a future sample of size 1000 for the pooled data, as requested with the PREDICT(1000) option. The "Sample Size for Estimation" table gives the sample size required to estimate the binomial probability to within 0.05 for the pooled data, as requested with the TOLERANCE(.05) option.

The "Estimates by Group" table supplies the estimated binomial probability, confidence limits, and the contribution to the total chi-squared for each board. The pooled values are shown on the last line of the table.

The "Predicted Values by Group" table gives the predicted counts in a future sample of size 1000, prediction limits, and the sample size required to estimate the binomial probability to within the tolerance of 0.05 for each board. Values for the pooled data are shown on the last line of the table.

The RELIABILITY Procedure

Model Information - All Groups

Input Data Set	WORK.BINEX
Events Variable	fail
Trials Variable	sample
Distribution	Binomial
Confidence Coefficient	95%
Observations Used	9

Binomial Data Analysis

Pooled Events	81.0000
Pooled Trials	2058.0000
Estimate of Proportion	0.0394
Lower Limit For Proportion	0.0314
Upper Limit For Proportion	0.0487
ChiSquare	56.8504
Pr>ChiSquare	0.0000

Predicted Value and Limits

Sample Size For Prediction	1000.0000
Predicted Count	39.3586
Lower Prediction Limit	24.8424
Upper Prediction Limit	56.3237

Sample Size For Estimation

Tolerance	0.0500
Sample Size For Tolerance	58.0975

Estimates By Group

				95% Confidence Limits		
Group	Events	Trials	Prop	Lower	Upper	X2
1	2	84	0.0238	0.0029	0.0834	0.5371
2	3	72	0.0417	0.0087	0.1170	0.0101
3	5	72	0.0694	0.0229	0.1547	1.7237
4	19	119	0.1597	0.0990	0.2381	45.5528
5	21	538	0.0390	0.0243	0.0590	0.0015
6	2	51	0.0392	0.0048	0.1346	0.0000
7	9	517	0.0174	0.0080	0.0328	6.5884
8	18	462	0.0390	0.0233	0.0609	0.0019
9	2	143	0.0140	0.0017	0.0496	2.4348
Pooled	81	2058	0.0394	0.0314	0.0487	56.8504

Predicted/Tolerance Values By Group

	Predicted	95% Prediction Limits		Tolerance
Group	Count	Lower	Upper	Sample Size
1	23.81	1.5476	88.5824	35.71
2	41.67	6.9416	124.6142	61.36
3	69.44	20.4052	165.3499	99.30
4	159.66	91.9722	254.5444	206.17
5	39.03	20.1599	64.7140	57.64
6	39.22	3.3970	144.2494	57.90
7	17.41	5.3506	36.7531	26.28
8	38.96	19.3343	66.3850	57.53
9	13.99	0.3851	53.0715	21.19
Pooled	39.36	24.8424	56.3237	58.10

Figure 37.41. Analysis of the Capacitor Data

Syntax

Primary Statements

The following are the primary statements that control the RELIABILITY procedure:

PROC RELIABILITY *<options>;*

*<label:>***ANALYZE** *variable<∗censor-variable(values)> <=(group-variables)> </ options>;*

*<label:>***MCFPLOT** *variable<∗cost/censor-variable(values)> <=(group-variables)> </ options>;*

MODEL *variable<∗censor-variable(values)> =<independent-variables> </ options>;*

*<label:>***PROBPLOT** *variable<∗censor-variable(values)> <=(group-variables)> </ options>;*

*<label:>***RELATIONPLOT** *variable<∗censor-variable(values)> <=(group-variables)> </ options>;*

The PROC RELIABILITY statement invokes the procedure.

The plot statements (PROBPLOT, RELATIONPLOT, and MCFPLOT) create graphical displays. Each of the plot statements has options that control the content and appearance of the plots they create. The default settings provide the best plots for many purposes; however, if you want to control specific details of the plots, such as axis limits or background colors, then you need to specify the options.

In addition to graphical output, each plot statement provides analysis results in tabular form. The tabular output also can be controlled with statement options.

The MODEL and ANALYZE statements produce only tabular analysis output, not graphical displays.

You can specify one or more of the plot and ANALYZE statements. If you specify more than one MODEL statement, only the last one specified is used.

Secondary Statements

You can specify the following statements in conjunction with the primary statements listed previously. These statements are used to modify the behavior of the primary statements or to specify additional variables.

BY *variables;*

CLASS *variables;*

DISTRIBUTION *distribution-name;*

FMODE *keyword = variable('value1' . . . 'valuen');*

FREQ *variable;*

INSET *keyword-list*< *options*>*;*

MAKE *'table'* **OUT=***SAS-data-set* < */ options*>*;*

NENTER *variable;*

UNITID *variable;*

The BY statement specifies variables in the input data set that are used for BY processing. A separate analysis is performed for each group of observations defined by the levels of the BY variables. The input data set must be sorted in order of the BY variables.
The CLASS statement specifies variables in the input data set that serve as *indicator*, *dummy*, or *classification* variables in the MODEL statement.

The DISTRIBUTION statement specifies a probability distribution name for those statements that require a probability distribution for proper operation (the ANALYZE, PROBPLOT, MODEL, and RELATIONPLOT statements). If you do not specify a distribution with the DISTRIBUTION statement, the normal distribution is used.

The FMODE statement specifies what failure-mode data to include in the analysis of data. Use this statement in conjunction with the ANALYZE, MODEL, PROBPLOT, or RELATIONPLOT statements.

The FREQ statement specifies a variable that provides frequency counts for each observation in the input data set.

The INSET statement specifies what information is printed in the inset box created by the PROBPLOT or MCFPLOT statements. The INSET statement also controls the appearance of the inset box.

The MAKE statement creates a SAS data set from any of the tables produced by the procedure. You specify a table and a SAS data set name for the data set you want to create. There is a unique table name that identifies each table printed; see the tables in the "MAKE Statement" section.

The NENTER statement specifies interval-censored data having a special structure; these data are called *readout* data. Use the NENTER statement in conjunction with the FREQ statement.

The UNITID statement specifies a variable in the input data set that is used to identify each individual unit in an MCFPLOT statement.

Graphical Enhancement Statements

You can use the TITLE, FOOTNOTE, and NOTE statements to enhance printed output. If you are creating plots, you can also use the LEGEND and SYMBOL statements to enhance your plots. For details, refer to *SAS/GRAPH Software: Reference* and the section for the plot statement that you are using.

PROC RELIABILITY Statement

PROC RELIABILITY < *options* >;

The PROC RELIABILITY statement invokes the procedure. You can specify the following *options*.

Table 37.4. PROC RELIABILITY Statement Options

option	*description*
DATA=*SAS-data-set*	specifies an input data set
GOUT=*graphics-catalog*	specifies a catalog for saving graphical output
NAMELEN=n	specifies the length of effect names in tables and output data sets to be n characters long, where n is a value between 20 and 200 characters. The default length is 20 characters.

ANALYZE Statement

<*label:*> **ANALYZE** *variable*<∗*censor-variable(values)*> <**=(***group-variables***)**> < **/** *options* >;
<*label:*> **ANALYZE** *(variable1 variable2)* <**=***group-variables*> <**/** *options*>;

<*label:*> **ANALYZE** *variable1***(***variable2***)** <**=(***group-variables***)**> < **/** *options* >;

You use the ANALYZE statement to estimate the parameters of the probability distribution specified in the DISTRIBUTION statement without producing any graphical output. The ANALYZE statement performs the same analysis as the PROBPLOT statement, but it does not produce any plots. In addition, you can use the ANALYZE statement to analyze data with the binomial and Poisson distributions. The third format for the preceding ANALYZE statement applies only to Poisson and binomial data. You can use any number of ANALYZE statements after a PROC RELIABILITY statement; each ANALYZE statement produces a separate analysis. You can specify an optional *label* to distinguish between multiple ANALYZE statements in the output.

You must specify one *variable*. If your data are right censored, you must specify a *censor-variable* and, in parentheses, the *values* of the *censor-variable* that correspond to censored data values.

If you are using the binomial or Poisson distributions, you must specify *variable1* to represent a binomial or Poisson count and *variable2* to provide an exposure measure for the Poisson distribution or the binomial sample size for the binomial distribution.

You can optionally specify one or two *group-variables*. The ANALYZE statement produces an analysis for each level combination of the *group-variable* values. The observations in a given level are referred to as a *cell*.

The elements of the ANALYZE statement are described as follows.

variable
represents the data for which an analysis is to be produced. A *variable* must be a numeric variable in the input data set.

censor-variable(values)
indicates which observations in the input data set are right censored. You specify the values of *censor-variable* that represent censored observations by placing those values in parentheses after the variable name. If your data are not right censored, then you omit the specification of *censor-variable*; otherwise, *censor-variable* must be a numeric variable in the input data set.

(*variable1 variable2)*
is another method of specifying the data. You can use this syntax in a situation where uncensored, interval-censored, left-censored and right-censored values occur in the same set of data. Table 37.23 on page 1153 shows how you use this syntax to specify different types of censoring by using combinations of missing and nonmissing values. See "Lognormal Analysis with Arbitrary Censoring" on page 1100 for an example of using this syntax to create a probability plot.

variable1
represents the count data for which a Poisson or binomial analysis is to be produced. A *variable1* must be a numeric variable in the input data set.

variable2
provides either an exposure measure for a Poisson analysis or a binomial number of trials for a binomial analysis. A *variable2* must be a numeric variable in the input data set.

group-variables
are one or two group variables. If no group variables are specified, a single analysis is produced. The *group-variables* can be numeric or character variables in the input data set.

Note that the parentheses surrounding the *group-variables* are needed only if two group variables are specified.

options
control the features of the analysis. All *options* are specified after a slash (/) in the ANALYZE statement.

Summary of Options

The following tables summarize the options available in the ANALYZE statement. You can specify one or more of these options to control the parameter estimation and provide optional analyses.

Table 37.5. Analysis Options for Distributions Other than Poisson or Binomial

Option	Option Description
CONFIDENCE=*number*	specifies the confidence coefficient for all confidence intervals. Specify a *number* between 0 and 1. The default value is 0.95

Table 37.5. Analysis Options for Distributions Other than Poisson or Binomial (continued)

Option	Option Description
CONVERGE=*number*	specifies the convergence criterion for maximum likelihood fit. See "Maximum Likelihood Estimation" on page 1188 for details.
CONVH=*number*	specifies the convergence criterion for the relative Hessian convergence criterion See "Maximum Likelihood Estimation" on page 1188 for details.
CORRB	requests parameter correlation matrix
COVB	requests parameter covariance matrix
FITTYPE | FIT=	specifies method of estimating distribution parameters
LSYX	-least squares fit to the probability plot. The probability axis is the dependent variable.
| LSXY	-least squares fit to the probability plot. The lifetime axis is the dependent variable.
| MLE	-maximum likelihood (default)
| NONE	-no fit is computed
| WEIBAYES <(CONFIDENCE|CONF=*number*)>	-Weibayes *number* is the confidence coefficient for the Weibayes fit and is between 0 and 1. The default is 0.95.
ITPRINT	requests iteration history for maximum likelihood fit
ITPRINTEM	requests iteration history for the Turnbull algorithm
LRCL	requests likelihood ratio confidence intervals for distribution parameters
LRCLPER	requests likelihood ratio confidence intervals for distribution percentiles
LOCATION=*number* < LINIT >	specifies fixed or initial value of location parameter
MAXIT=*number*	specifies maximum number of iterations allowed for maximum likelihood fit
MAXITEREM | MAXITEM=*number1* <,*number2*>	*number1* specifies maximum number of iterations allowed for Turnbull algorithm. Iteration history will be printed in increments of *number2* if requested with ITPRINTEM. See "Interval-Censored Data" on page 1181 for details.
NOPCTILES	suppress computation of percentiles

Table 37.5. Analysis Options for Distributions Other than Poisson or Binomial (continued)

Option	Option Description
NOPOLISH	suppress setting small interval probabilities to zero in Turnbull algorithm. See "Interval-Censored Data" on page 1181 for details.
PCTLIST=*number-list*	specifies list of percentages for which to compute percentile estimates. *number-list* must be a list of numbers separated by blanks or commas. Each number in the list must be between 0 and 100
PPOS=	specifies plotting position type. See "Probability Plotting" beginning on page page 1177 for details.
EXPRANK | MEDRANK | MEDRANK1 | KM | MKM | (NA | NELSONAALEN)	-expected ranks -median ranks -median ranks (exact formula) -Kaplan-Meier -modified Kaplan-Meier (default) - Nelson-Aalen
PPOUT	request table of cumulative probabilities
PRINTPROBS	print intervals and associated probabilities for the Turnbull algorithm
PROBLIST=*number-list*	specifies list of initial values for Turnbull algorithm. See "Interval-Censored Data" on page 1181 for details.
PSTABLE=*number*	specifies stable parameterization. The *number* must be between zero and one. See "Stable Parameters" on page 1192 for further information.
READOUT	analyze readout data
SCALE=*number* < SCINIT >	specifies fixed or initial value of scale parameter
SHAPE=*number* < SHINIT >	specifies fixed or initial value of shape parameter
SINGULAR=*number*	specifies singularity criterion for matrix inversion
SURVTIME=*number-list*	requests survival function be computed for values in *number-list*
THRESHOLD=*number*	specifies a fixed threshold parameter. See Table 37.40 for the distributions with a threshold parameter.
TOLLIKE=*number*	specifies criterion for convergence in the Turnbull algorithm. Default is 10^{-8}. See "Interval-Censored Data" on page 1181 for details.

Table 37.5. Analysis Options for Distributions Other than Poisson or Binomial (continued)

Option	Option Description
TOLPROB=*number*	specifies criterion for setting interval probability to zero in the Turnbull algorithm. Default is 10^{-6}. See "Interval-Censored Data" on page 1181 for details.

Table 37.6. Analysis Options for Poisson And Binomial Distributions

Option	Option Description
CONFIDENCE=*number*	specifies the confidence coefficient for all confidence intervals. Specify a *number* between 0 and 1. The default value is 0.95
PREDICT(*number*)	requests predicted counts for exposure *number* for Poisson or sample size *number* for binomial
TOLERANCE(*number*)	requests exposure for Poisson or sample size for binomial to estimate Poisson rate or binomial probability within *number* with probability given by the CONFIDENCE= option

BY Statement

BY *variable-names*

You can specify a BY statement with PROC RELIABILITY to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

CLASS Statement

CLASS *variable-names*

The CLASS statement specifies variables in the input data set that serve as *indicator*, *dummy*, or *classification* variables in the MODEL statement. If a CLASS variable is specified as an independent variable in the MODEL statement, the RELIABILITY procedure automatically generates an indicator variable for each level of the CLASS variable. The indicator variables generated are used as independent variables in the regression model specified in the MODEL statement. An indicator variable for a level of a CLASS variable is a variable equal to 1 for those observations corresponding to the level and equal to 0 for all other observations.

DISTRIBUTION Statement

DISTRIBUTION *probability distribution-name*

The ANALYZE, PROBPLOT, RELATIONPLOT, and MODEL statements require

you to specify the probability distribution that describes your data. You can specify a probability distribution using the DISTRIBUTION statement anywhere after the PROC RELIABILITY statement and before the RUN statement. If you do not specify a distribution in a DISTRIBUTION statement, the normal distribution is assumed. The probability distribution specified determines the distribution for which parameters are estimated using your data. The valid distributions and the statements to which they apply are shown in Table 37.7.

Table 37.7. Probability Distributions

Distribution	Distribution-Name Specified	Statement
binomial	BINOMIAL	ANALYZE
exponential	EXPONENTIAL	ANALYZE, PROBPLOT, RELATIONPLOT, MODEL
extreme value	EXTREME | EV	ANALYZE, PROBPLOT, RELATIONPLOT, MODEL
generalized gamma	GAMMA	MODEL
logistic	LOGISTIC | LOGIT	ANALYZE, PROBPLOT, RELATIONPLOT, MODEL
loglogistic	LLOGISTIC | LLOGIT	ANALYZE, PROBPLOT, RELATIONPLOT, MODEL
lognormal (base e)	LOGNORMAL | LNORM	ANALYZE, PROBPLOT, RELATIONPLOT, MODEL
lognormal (base 10)	LOGNORMAL10 | LNORM10	ANALYZE, PROBPLOT, RELATIONPLOT, MODEL
normal	NORMAL	ANALYZE, PROBPLOT, RELATIONPLOT, MODEL
Poisson	POISSON	ANALYZE
Weibull	WEIBULL	ANALYZE, PROBPLOT, RELATIONPLOT, MODEL

FMODE Statement

FMODE *keyword*= *variable* (*'value1' . . . 'valuen'*) < **/** *options* >**;**

Use the FMODE statement with data that have failures attributable to multiple causes, or *failure modes*. You can analyze data by either keeping, eliminating, or combining specific failure modes with the FMODE statement. Use this statement with the KEEP or ELIMINATE keywords in conjunction with the ANALYZE, MODEL, PROBPLOT, or RELATIONPLOT statements. Use this statement with the COMBINE keyword with the ANALYZE or PROBPLOT statements. You can place an FMODE statement anywhere after the PROC RELIABILITY statement and before the RUN statement.

If you specify the keyword KEEP, the life distribution for only the identified failure modes is estimated, with all other failure modes treated as right-censored data. If you specify the keyword ELIMINATE, the life distribution that results if the failure modes identified are completely eliminated is estimated. The keyword ELIMINATE causes the failure modes identified to be treated as right-censored data and causes a single life distribution to be estimated for the remaining data. If you specify the

keyword COMBINE, the data are analyzed with all the specified failure modes combined acting. See "Weibull Probability Plot for Two Combined Failure Modes" on page 1115 for an example of a Weibull plot of data with two combined failure modes. The failure mode for an observation in the input data set is identified by the value of *variable*, where *variable* is any numeric or character variable in the input data set. You must identify a failure mode for each observation that is not right-censored. You specify failure modes to keep, eliminate, or combine by listing variable-values *(value1 ... valuen)* in parentheses after the failure mode variable name. The list of variable-values must have entries separated by blanks or commas. You can specify the following *options* after the slash (/). These options will affect the analysis only when you use the COMBINE keyword.

Table 37.8. FMODE Statement Options

option	*description*
LEGEND=	specifies a LEGEND statement for individual mode fit lines
NOLEGEND	suppress legend for individual mode fit lines
PLOTMODES	plot individual failure distribution lines on probability plot

FREQ Statement

FREQ *variable-name*

The FREQ statement specifies a variable that provides frequency counts for each observation in the input data set. If n is the value of the FREQ variable for an observation, then that observation is weighted by n. The log-likelihood function for maximum likelihood estimation is multiplied by n. If n is not an integer, the integer part of n is used in creating probability plots.

You can also use the FREQ statement in conjunction with the NENTER statement to specify interval-censored data having a special structure; these data are called *readout* data. The FREQ statement specifies a variable in the input data set that determines the number of units failing in each interval. See "Weibull Analysis of Interval Data with Common Inspection Schedule" on page 1096 for an example using the FREQ statement with readout data.

You can use the FREQ statement with the MCFPLOT statement for exact age data to provide frequency counts for entire recurrence histories. If n is the value of the FREQ variable at a censor time, the history of recurrences for the corresponding system is replicated independently n times. Values of the FREQ variable at times other than censor times are not used, and may be any value or missing without affecting the analysis. The FREQ variable is not used with the MCFPLOT statement for interval recurrence data, and the values of a FREQ variable specified in this case are ignored.

INSET Statement

INSET *keyword-list*< **/** *options* >**;**

The box or table of summary information produced on plots made with the PROBPLOT or MCFPLOT statement is called an *inset*. You can use the INSET statement to customize the information that is printed in the inset box and the appearance of the inset box. To supply the information that is displayed in the inset box, you specify *keywords* corresponding to the information you want shown. For example, the following statements produce a Weibull plot with the sample size, the number of failures, and the Weibull mean displayed in the inset.

```
proc reliability data=fan;
   distribution weibull;
   pplot lifetime*censor(1);
   inset n nfail weibull(mean);
run;
```

By default, inset entries are identified with appropriate labels. However, you can provide a customized label by specifying the *keyword* for that entry followed by the equal sign (=) and the label in quotes. For example, the following INSET statement produces an inset containing the sample size and Weibull mean, labeled “Sample Size” and “Weibull Mean” in the inset.

```
inset n='Sample Size' weibull(mean='Weibull Mean');
```

If you specify a keyword that does not apply to the plot you are creating, then the keyword is ignored.

The *options* control the appearance of the box.

If you specify more than one INSET statement, only the last one is used.

Keywords Used in the INSET Statement

The following tables list keywords available in the INSET statement to display summary statistics, distribution parameters, and distribution fitting information.

Table 37.9. Summary Statistics

keyword	*description*
N	sample size
NFAIL	number of failures for probability plots
NEVENTS	number of events or repairs for MCF plots
NEVENTS1	number of events or repairs in the first group for MCF difference plots
NEVENTS2	number of events or repairs in the second group for MCF difference plots
NUNITS	number of units or systems for MCF plots
NUNITS1	number of units or systems in the first group for MCF difference plots
NUNITS2	number of units or systems in the second group for MCF difference plots

Table 37.10. General Information

keyword	*description*
CONFIDENCE	confidence coefficient for all confidence intervals or for the Weibayes fit
FIT	method used to estimate distribution parameters for probability plots
RSQUARE	R^2 for least squares distribution fit to probability plots

Distribution parameters are specified as *distribution-name*(*distribution-parameters*). The following table lists the keywords available.

Table 37.11. Distribution Parameters

keyword	*secondary keyword*	*description*
EXPONENTIAL	SCALE	scale parameter
	THRESHOLD	threshold parameter
	MEAN	expected value
EXTREME | EV	LOCATION	location parameter
	SCALE	scale parameter
	MEAN	expected value
LOGISTIC | LOGIT	LOCATION	location parameter
	SCALE	scale parameter
	MEAN	expected value
LOGLOGISTIC | LLOGIT	LOCATION	location parameter
	SCALE	scale parameter
	THRESHOLD	threshold parameter

Table 37.11. Distribution Parameters (continued)

keyword	*secondary keyword*	*description*
	MEAN	expected value
LOGNORMAL	LOCATION	location parameter
	SCALE	scale parameter
	THRESHOLD	threshold parameter
	MEAN	expected value
LOGNORMAL10	LOCATION	location parameter
	SCALE	scale parameter
	THRESHOLD	threshold parameter
	MEAN	expected value
NORMAL	LOCATION	location parameter
	SCALE	scale parameter
	MEAN	expected value
WEIBULL	SCALE	scale parameter
	SHAPE	shape parameter
	THRESHOLD	threshold parameter
	MEAN	expected value

Options Used in the INSET Statement

The following tables list INSET statement options that control the appearance of the inset box.

Table 37.12. General Appearance Options

Option	**Option Description**
HEADER=*'quoted string'*	specifies text for header or box title
NOFRAME	omits frame around box
POS= *value* <DATA | PERCENT>	determines the position of the inset. The *value* can be a compass point (N, NE, E, SE, S, SW, W, NW) or a pair of coordinates (x, y) enclosed in parentheses. The coordinates can be specified in axis percent units or axis data units.
REFPOINT= *name*	specifies the reference point for an inset that is positioned by a pair of coordinates with the POS= option. You use the REFPOINT= option in conjunction with the POS= coordinates. The REFPOINT= option specifies which corner of the inset frame you have specified with coordinates (x, y) and it can take the value of BR (bottom right), BL (bottom left), TR (top right), or TL (top left). The default is REFPOINT=BL. If the inset position is specified as a compass point, then the REFPOINT= option is ignored.

Table 37.13. Text Enhancement Options

Option	Option Description
FONT=*font*	software font for text
HEIGHT=*value*	height of text

Table 37.14. Color and Pattern Options

Option	Option Description
CFILL=*color*	color for filling box
CFILLH=*color*	color for filling box header
CFRAME=*color*	color for frame
CHEADER=*color*	color for text in header
CTEXT=*color*	color for text

LOGSCALE Statement

LOGSCALE <*effect-list*> < **/** *options* >**;**

You use the LOGSCALE statement to model the logarithm of the distribution scale parameter as a function of explanatory variables. A MODEL statement must also be present to specify the model for the distribution location parameter. *effect-list* is a list of variables in the input data set representing the values of the independent variables in the model for each observation, and combinations of variables representing interaction terms. It can contain any variables or combination of variables in the input data set. It can contain the same variables as the MODEL statement, or it can contain different variables. The variables in the *effect-list* can be any combination of indicator variables named in a CLASS statement and continuous variables. The coefficients of the explanatory variables are estimated by maximum likelihood.

The following *options* are available for the LOGSCALE statement.

Table 37.15. LOGSCALE Statement Options

Option	Option Description
INITIAL=*number list*	specifies initial values for log-scale regression parameters other than the intercept term
INTERCEPT=*number* < INTINIT >	specifies initial or fixed value of the intercept parameter, depending on whether INTINIT is present

MAKE Statement

MAKE *'table'* **OUT=***SAS-data-set*<*(SAS-data-set options)*>**;**

The MAKE statement creates a SAS data set from any of the tables produced by the RELIABILITY procedure. You can specify SAS data set options in parentheses after

the data set name. You can specify one MAKE statement for each table that you want to save to a SAS data set.

The ODS statement also creates SAS data sets from tables, in addition to providing an extensive and flexible method of controlling output created by the RELIABILITY procedure. The ODS statement is the recommended method of controlling procedure output, however, the MAKE statement is provided for compatibility with earlier releases of the SAS system

The valid values for *table* are shown in "ODS Table Names" on page 1213, organized by the RELIABILITY procedure statement that produces the tabular output. The *table* names are not case sensitive, but they must be enclosed in single quotes.

MCFPLOT Statement

*<label:>***MCFPLOT** *variable ∗cost/censor-variable(values) <***=(***group-variables***)***> <***/***options>***;***
*<label:>***MCFPLOT** *(<***INTERVAL=***> variable1 variable2*

*<***RECURRENCES=***> variable3*
*<***CENSOR=***> variable4) <***=(***group-variables***)***> <***/***options>***;***

You can specify any number of MCFPLOT statements after a PROC RELIABILITY statement. Each MCFPLOT statement creates a separate MCF plot and associated analysis. See "Analysis of Recurrence Data on Repairs" on page 1118, "Comparison of Two Samples of Repair Data" on page 1122, and "Analysis of Interval Age Recurrence Data" on page 1126 for examples using the MCFPLOT statement. You can specify an optional *label* to distinguish between multiple MCFPLOT statements in the output.

To create a plot of the mean cumulative function for cost or number of repairs with exact age data, you specify a *variable* that represents the times of repairs. You must also specify a *cost/censor-variable* and the *values*, in parentheses, of the *cost/censor-variable* that correspond to end-of-history data values (also referred to as *censored* data values).

To create a plot of the mean cumulative function for cost or number of repairs with interval age data, you specify *variable1 variable2* that represents the age intervals, *variable3* that represents the number of recurrences in the intervals, and *variable4* that represents the number censored in the intervals.

You can optionally specify one or two *group-variables* (also referred to as *classification variables*). The MCFPLOT statement displays a component plot for each level of the *group-variables* using the values of the *variable*. The observations in a given level are referred to as a *cell*.

For exact data, you must also specify a *unit-identification* variable in conjunction with the MCFPLOT statement to identify the individual unit name for each instance of repair or end of history on the unit. Specify the *unit-identification* variable in the UNITID statement.

The elements of the MCFPLOT statement are described as follows.

variable
represents the time of repair. A *variable* must be a numeric variable in the input data set.

variable1 variable2
represents time intervals for grouped data. *variable1* and *variable2* must be numeric variables in the input data set.

variable3
represents the number of recurrences in an interval. A *variable3* must be a numeric variable in the input data set.

variable4
represents the number censored in an interval. A *variable4* must be a numeric variable in the input data set.

cost/censor-variable(values)
indicates the cost of each repair or the number of repairs. This variable also indicates which observations in the input data set are end-of-history (censored) data points. You specify the values of *cost/censor-variable* that represent censored observations by placing those values in parentheses after the variable name. A *censor-variable* must be a numeric variable in the input data set.

group-variables
are one or two group variables. If no group variables are specified, a single plot is produced. The *group-variables* can be any numeric or character variables in the input data set.

Note that the parentheses surrounding the *group-variables* are needed only if two group variables are specified.

options
control the features of the mean cumulative function plot. All *options* are specified after a slash (/) in the MCFPLOT statement. The "Summary of Options" section, which follows, lists all options by function.

Summary of Options

Table 37.16. Analysis Options

Option	Option Description
CONFIDENCE=*number*	specifies the confidence coefficient for all confidence intervals. Specify a *number* between 0 and 1. The default value is 0.95
INDINC	requests variance estimates of the MCF computed using the Nelson estimator under an independent increments assumption

Table 37.16. Analysis Options (continued)

Option	Option Description
LOGINTERVALS	requests confidence intervals be computed based on the asymptotic normality of log(MCF). This is appropriate only when the MCF estimate is positive, so does not apply to MCF differences or costs if costs can be negative.
MCFDIFF	requests a plot of differences of MCFS of two groups specified by a single group variable
NOVARIANCE	suppresses MCF variance computation
VARIANCE=	alternate method of specifying variance calculation. Includes INDINC and VARMETHOD2 options.
INDINC	Nelson's method assuming independent increments
| LAWLESS	Lawless-Nadeau method
| NELSON	Nelson's method (the default method)
| POISSON	Poisson process method
VARMETHOD2	requests the method of Lawless and Nadeau (1995) be used to compute variance estimates of the MCF

Table 37.17. Plot Layout Options

Option	Option Description
CENBIN	plots censored data as frequency counts rather than as individual points
CENSYMBOL=*symbol* | (*symbol list*)	specifies symbols for censored values. *symbol* is one of the symbol names (plus, star, square, diamond, triangle, hash, paw, point, dot, circle) or a letter (A–Z). If you are creating overlaid plots for groups of data, you can specify different symbols for the groups with a list of symbols or letters, separated by blanks, enclosed in parentheses. If no CENSYMBOL option is specified, the symbol used for censored values is the same as for repairs.
HOFFSET=*value*	specifies offset for horizontal axis
INBORDER	requests a border around MCF plots
INTERPOLATE=JOIN | STEP	requests symbols in an MCF plot be connected with a straight line or step function
INTERTILE=*value*	specifies distance between tiles
MCFLEGEND=*legend-statement-name* | NONE	identifies legend statement to specify legend for overlaid MCF plots

Table 37.17. Plot Layout Options (continued)

Option	Option Description
MISSING1	requests that missing values of first GROUP= variable be treated as a level of the variable
MISSING2	requests that missing values of second GROUP= variable be treated as a level of the variable
NCOLS=n	specifies number of columns plotted on a page
NOCENPLOT	suppresses plotting of censored data points
NOCONF	suppresses plotting of confidence intervals
NOFRAME	suppresses frame around plotting area
NOINSET	suppresses inset
NOLEGEND	suppresses legend for overlaid MCF plots
NROWS=n	specifies number of rows plotted on a page
ORDER1=DATA | FORMATTED | FREQ | INTERNAL	specifies display order for values of the first GROUP= variable
ORDER2=DATA | FORMATTED | FREQ | INTERNAL	specifies display order for values of the second GROUP= variable
OVERLAY	requests plots with group variables be overlaid on a single page
PLOTSYMBOL=*symbol* | (*symbol list*)	symbols representing events in an MCF plot
PLOTCOLOR=*color* | (*color list*)	colors of symbols representing events in an MCF plot
TURNVLABELS	vertically strings out characters in labels for vertical axis
VOFFSET=*value*	specifies length of offset at upper end of vertical axis

Table 37.18. Reference Line Options

Option	Option Description
HREF=*value-list*	specifies reference lines perpendicular to horizontal axis
HREFLABELS=*('label1' . . . 'labeln')*	specifies labels for HREF= lines.

Table 37.18. Reference Line Options (continued)

Option	Option Description
HREFLABPOS=*n*	specifies vertical position of labels for HREF= lines. The valid values for *n* and the corresponding label placements are shown below. *n* | **label placement** 1 | top 2 | staggered from top 3 | bottom 4 | staggered from bottom 5 | alternating from top 6 | alternating from bottom
LHREF=*linetype*	specifies line style for HREF= lines
LVREF=*linetype*	specifies line style for VREF= lines
VREF=*value-list*	specifies reference lines perpendicular to vertical axis
VREFLABELS=*('label1' . . . 'labeln')*	specifies labels for VREF= lines
VREFLABPOS=*n*	specifies horizontal position of labels for VREF= lines. The valid values for *n* and the corresponding label placements are shown below. *n* | **label placement** 1 | left 2 | right

Table 37.19. Text Enhancement Options

Option	Option Description
FONT=*font*	software font for text
HEIGHT=*value*	height of text used outside framed areas
INFONT=*font*	software font for text inside framed areas
INHEIGHT=*value*	height of text inside framed areas

Table 37.20. Axis Options

Option	Option Description

Table 37.20. Axis Options (continued)

Option	Option Description

HAXIS=*value1 to value2<by value3>*	specifies tick mark values for the horizontal axis. *value1*, *value2*, and *value3* must be numeric, and *value1* must be less than *value2*. The lower tick mark is *value1*. Tick marks are drawn at increments of *value3*. The last tick mark is the greatest value that does not exceed *value2*. If *value3* is omitted, a value of 1 is used. This method of specification of tick marks is not valid for logarithmic axes. Examples of HAXIS= lists follow:
	`haxis = 0 to 10`
	`haxis = 2 to 10 by 2`
	`haxis = 0 to 200 by 10`

HLOWER=*number*	specifies the lower limit on the horizontal axis scale. The HLOWER= option specifies *number* as the lower horizontal axis tick mark. The tick mark interval and the upper axis limit are determined automatically. This option has no effect if the HAXIS option is used.
HUPPER=*number*	specifies the upper limit on the horizontal axis scale. The HUPPER= option specifies *number* as the upper horizontal axis tick mark. The tick mark interval and the lower axis limit are determined automatically. This option has no effect if the HAXIS= option is used.
LGRID=*number*	specifies a line style for all grid lines. *number* is between 1 and 46 and specifies a linestyle for grids.
LOGLOG	requests log scales on both axes
MINORLOGGRID	adds a minor grid for log axes
NOGRID	suppresses grid lines
NOHLABEL	suppresses label for horizontal axis
NOVLABEL	suppresses label for vertical axis
NOVTICK	suppresses tick marks and tick mark labels for vertical axis
NOHTICK	suppresses tick marks and tick mark labels for horizontal axis
NHTICK=*number*	specifies number of tick intervals for the horizontal axis. This option has no effect if the HAXIS= option is used.

Table 37.20. Axis Options (continued)

Option	Option Description
NVTICK=*number*	specifies number of tick intervals for the vertical axis. This option has no effect if the VAXIS= option is used.

VAXIS=*value1 to value2<by value3>*	specifies tick mark values for the vertical axis. *value1*, *value2*, and *value3* must be numeric, and *value1* must be less than *value2*. The lower tick mark is *value1*. Tick marks are drawn at increments of *value3*. The last tick mark is the greatest value that does not exceed *value2*. This method of specification of tick marks is not valid for logarithmic axes. If *value3* is omitted, a value of 1 is used.
	`vaxis = 0 to 10`
	`vaxis = 0 to 2 by .1`

VAXISLABEL='*string*'	specifies a label for the vertical axis
VLOWER=*number*	specifies the lower limit on the vertical axis scale. The VLOWER= option specifies *number* as the lower vertical axis tick mark. The tick mark interval and the upper axis limit are determined automatically. This option has no effect if the VAXIS= option is used.
VUPPER=*number*	specifies the upper limit on the vertical axis scale. The VUPPER= option specifies *number* as the upper vertical axis tick mark. The tick mark interval and the lower axis limit are determined automatically. This option has no effect if the VAXIS= option is used.
WAXIS=n	specifies line thickness for axes and frame

Table 37.21. Color and Pattern Options

Option	Option Description
CAXIS=*color*	color for axis
CCENSOR=*color*	color for filling censor plot area
CENCOLOR=*color*	color for censor symbol
CFRAME=*color*	color for frame
CFRAMESIDE=*color*	color for filling frame for row labels
CFRAMETOP=*color*	color for filling frame for column labels
CGRID=*color*	color for grid lines
CHREF=*color*	color for HREF= lines
CTEXT=*color*	color for text
CVREF=*color*	color for VREF= lines

Table 37.22. Graphics Catalog Options

Option	Option Description
DESCRIPTION='*string*'	description for graphics catalog member
NAME='*string*'	name for plot in graphics catalog

MODEL Statement

MODEL *variable*<∗*censor-variable(values)*> <**=***effect-list*> < **/** *options* >**;**
MODEL *(variable1 variable2)* <**=***effect-list*> <**/***options*>**;**

You use the MODEL statement to fit regression models, where life is modeled as a function of explanatory variables.

You can use only one MODEL statement after a PROC RELIABILITY statement. If you specify more than one MODEL statement, only the last is used.

The MODEL statement does not produce any plots, but it enables you to analyze more complicated regression models than the ANALYZE, PROBPLOT, or RELATIONPLOT statement does. The probability distribution specified in the DISTRIBUTION statement is used in the analysis. The following are examples of MODEL statements:

```
model time = temp voltage;
model life*censor(1) = voltage width;
```

See "Analysis of Accelerated Life Test Data" on page 1090 and "Regression Modeling" on page 1104 for examples of fitting regression models using the MODEL statement.

If your data are right censored, you must specify a *censor-variable* and, in parentheses, the *values* of the *censor-variable* that correspond to censored data values.

If your data contain any interval-censored or left-censored values, you must specify *variable1* and *variable2* in parentheses to provide the endpoints of the interval for each observation.

The independent variables in your regression model are specified in the *effect-list*. The *effect-list* is any combination of continuous variables, classification variables,

See Regression Models on page 1190 for further information on specifying the independent variables.

The elements of the MODEL statement are described as follows.

variable
is the dependent, or response, variable. The *variable* must be a numeric variable in the input data set.

censor-variable(values)
indicates which observations in the input data set are right censored. You specify the values of *censor-variable* that represent censored observations by placing those

values in parentheses after the variable name. If your data are not right censored, then you can omit the specification of a *censor-variable*; otherwise, *censor-variable* must be a numeric variable in the input data set.

(variable1 variable2)
is another method of specifying the dependent variable in the regession model. You can use this syntax in a situation where uncensored, interval-censored, left-censored and right-censored values occur in the same set of data. Table 37.23 shows how you use this syntax to specify different types of censoring by using combinations of missing and nonmissing values.

Table 37.23. Specifying Censored Values

Variable1	Variable2	Type of Censoring
nonmissing	nonmissing	uncensored if *variable1* = *variable2*
nonmissing	nonmissing	interval censored if *variable1* < *variable2*
nonmissing	missing	right censored at *variable1*
missing	nonmissing	left censored at *variable2*

For example, if T1 and T2 represent time in hours in the input data set

```
OBS     T1     T2

 1       .      6
 2       6     12
 3      12     24
 4      24      .
 5      24     24
```

then the statement

```
model (t1 t2);
```

specifies a model in which observation 1 is left censored at 6 hours, observation 2 is interval censored in the interval (6, 12), observation 3 is interval censored in (12,24), observation 4 is right censored at 24 hours, and observation 5 is an uncensored lifetime of 24 hours.

effect-list
is a list of variables in the input data set representing the values of the independent variables in the model for each observation, and combinations of variables representing interaction terms. If a variable in the *effect-list* is also listed in a CLASS statement, an indicator variable is generated for each level of the variable. An indicator variable for a particular level is equal to 1 for observations with that level, and equal to 0 for all other observations. This type of variable is called a *classification* variable. Classification variables can be either character or numeric. If a variable is not listed in a CLASS statement, it is assumed to be a continuous variable, and it must be numeric.

options
control how the model is fit and what output is produced. All *options* are specified after a slash (/) in the MODEL statement. The "Summary of Options" section, which follows, lists all options by function.

Summary of Options

Table 37.24. Model Statement Options

Option	Option Description
CONFIDENCE=*number*	specifies the confidence coefficient for all confidence intervals. Specify a *number* between 0 and 1. The default value is 0.95
CONVERGE=*number*	specifies the convergence criterion for maximum likelihood fit. See "Maximum Likelihood Estimation" on page 1188 for details.
CONVH=*number*	specifies the convergence criterion for the relative Hessian convergence criterion See "Maximum Likelihood Estimation" on page 1188 for details.
CORRB	requests parameter correlation matrix
COVB	requests parameter covariance matrix
INITIAL=*number list*	specifies initial values for regression parameters other than the location, or intercept term
ITPRINT	requests iteration history for maximum likelihood fit
LRCL	requests likelihood ratio confidence intervals for distribution parameters
LOCATION=*number* < LINIT >	specifies fixed or initial value of the location, or intercept parameter
MAXIT=*number*	specifies maximum number of iterations allowed for maximum likelihood fit
OBSTATS	requests a table containing the XBETA, SURV, SRESID, and ADJRESID statistics in Table 37.25. The table also contains the dependent and independent variables in the model.
OBSTATS(*statistics*)	requests a table containing the model variables and the statistics in the specified list of *statistics*. Available statistics are shown in Table 37.25.
ORDER=DATA | FORMATTED | FREQ | INTERNAL	specifies sort order for values of the classification variables in the *effect-list*
PSTABLE=*number*	specifies stable parameterization. The *number* must be between zero and one. See "Stable Parameters" on page 1192 for further information.

Table 37.24. Model Statement Options (continued)

Option	Option Description
READOUT	analyzes data in readout structure. The FREQ statement must be used to specify the number of units failing in each interval, and the NENTER statement must be used to specify the number of unfailed units entering each interval
RELATION=ARRHENIUS | ARRHENIUS2 | POWER | LINEAR | LOGISTIC RELATION=(ARRHENIUS | ARRHENIUS2 | POWER | LINEAR | LOGISTIC < , > ARRHENIUS | ARRHENIUS2 | POWER | LINEAR | LOGISTIC)	specifies type of relationship between independent and dependent variables. In the first form, the transformation specified is applied to the first continuous independent variable in the model. In the second form, the transformations specified within parentheses are applied to the first two continuous independent variables in the model, in the order listed. See Table 37.49 on page 1191 for definitions of the transformations.
SCALE=*number* < SCINIT >	specifies fixed or initial value of scale parameter
SHAPE=*number* < SHINIT >	specifies fixed or initial value of shape parameter
SINGULAR=*number*	specifies singularity criterion for matrix inversion
THRESHOLD=*number*	specifies a fixed threshold parameter. See Table 37.40 for the distributions with a threshold parameter.

Table 37.25. Observation Statistics Available in the OBSTATS Option

Option	Option Description
CENSOR	is an indicator variable equal to 1 if an observation is censored, and 0 otherwise
QUANTILES | QUANTILE | Q=*number list*	specifies distribution quantiles for each number in *number list* for each observation. The numbers must be between 0 and 1. Estimated quantile standard errors, and upper and lower confidence limits are also tabulated.
XBETA	is the linear predictor

Table 37.25. Observation Statistics Available in the OBSTATS Option (continued)

Option	Option Description
SURVIVAL | SURV	is the fitted survival function, evaluated at the value of the dependent variable
RESID	is the raw residual
SRESID	is the standardized residual
GRESID	is the modified Cox-Snell residual
DRESID	is the deviance residual
ADJRESID	is the adjusted standardized residuals. These are adjusted for right-censored observations by adding the median of the lifetime greater than the right-censored values to the residuals.
RESIDADJ=*number*	specifies adjustment to be added to Cox-Snell residual for right-censored data values. The default is log(2) = 0.693.
RESIDALPHA | RALPHA=*number*	specifies $number \times 100\%$ percentile residual lifetime used to adjust right-censored standardized residuals. The *number* must be between 0 and 1. The default value is 0.5, corresponding to the median.
CONTROL=*variable*	specifies a control variable in the input data set. If the value of the control variable is 1, the observation statistics are computed. If the value of the control variable is not equal to 1, the statistics are not computed for that observation.

NENTER Statement

NENTER *variable*

Use the NENTER statement in conjunction with the FREQ statement to specify interval-censored data having a special structure; these data are called *readout* data. The NENTER statement specifies a *variable* in the input data set that determines the number of unfailed units entering each interval. See "Weibull Analysis of Interval Data with Common Inspection Schedule" on page 1096 for an example using the NENTER statement with readout data.

PROBPLOT Statement

*<label:>***PROBPLOT** *variable<∗censor-variable(values)> <***=(***group-variables***)***> <***/***options***>***;*
*<label:>***PROBPLOT** *(variable1 variable2) <***=***group-variables***> <***/***options***>***;*

You use the PROBPLOT statement to create a probability plot from complete, left-censored, right-censored, or interval censored data.

You can specify the keyword PPLOT as an alias for PROBPLOT. You can specify any number of PROBPLOT statements after a PROC RELIABILITY statement. Each PROBPLOT statement creates a probability plot and an associated analysis. The probability distribution used in creating the probability plot and performing the analysis is determined by the DISTRIBUTION statement. You can specify an optional *label* to distinguish between multiple PROBPLOT statements in the output.

See "Analysis of Right-Censored Data from a Single Population" on page 1085 and "Weibull Analysis Comparing Groups of Data" on page 1088 for examples creating probability plots using the PROBPLOT statement.

To create a probability plot, you must specify one *variable*. If your data are right censored, you must specify a *censor-variable* and, in parentheses, the *values* of the *censor-variable* that correspond to censored data values.

You can optionally specify one or two *group-variables* (also referred to as *classification variables*). The PROBPLOT statement displays a component probability plot for each level of the *group-variables* using the values of the *variable*. The observations in a given level are referred to as a *cell*.

The elements of the PROBPLOT statement are described as follows.

variable
represents the data for which a probability plot is to be produced. The *variable* must be a numeric variable in the input data set.

censor-variable(values)
indicates which observations in the input data set are right censored. You specify the values of *censor-variable* that represent censored observations by placing those values in parentheses after the variable name. If your data are not right censored, then you can omit the specification of *censor-variable*; otherwise, *censor-variable* must be a numeric variable in the input data set.

(variable1 variable2)
is another method of specifying the data for which a probability plot is to be produced. You can use this syntax in a situation where uncensored, interval-censored, left-censored and right-censored values occur in the same set of data. Table 37.23 on page 1153 shows how you use this syntax to specify different types of censoring by using combinations of missing and nonmissing values. See "Lognormal Analysis with Arbitrary Censoring" on page 1100 for an example of using this syntax to create a probability plot.

group-variables
are one or two group variables. If no group variables are specified, a single probability plot is produced. The *group-variables* can be numeric or character variables in the input data set.

Note that the parentheses surrounding the *group-variables* are needed only if two group variables are specified.

options
control the features of the probability plot. All *options* are specified after the slash (/) in the PROBPLOT statement. The "Summary of Options" section on page 1158, which follows, lists all options by function.

Summary of Options

Table 37.26. Analysis Options

Option	Option Description
CONFIDENCE=*number*	specifies the confidence coefficient for all confidence intervals. The *number* must be between 0 and 1. The default value is 0.95
CONVERGE=*number*	specifies the convergence criterion for maximum likelihood fit. See "Maximum Likelihood Estimation" on page 1188 for details.
CONVH=*number*	specifies the convergence criterion for the relative Hessian convergence criterion See "Maximum Likelihood Estimation" on page 1188 for details.
CORRB	requests parameter correlation matrix
COVB	requests parameter covariance matrix
FITTYPE | FIT=	specifies method of estimating distribution parameters
LSYX	-least squares fit to the probability plot. The probability axis is the dependent variable.
| LSXY	-least squares fit to the probability plot. The lifetime axis is the dependent variable.
| MLE	-maximum likelihood (default)
| MODEL	-use the fit from the preceding MODEL statement
| NONE	-no fit is computed
| WEIBAYES <(CONFIDENCE| CONF=*number*)>	-Weibayes method *number* is the confidence coefficient for the Weibayes fit and is between 0 and 1. The default value is 0.95.
ITPRINT	requests iteration history for maximum likelihood fit
ITPRINTEM	requests iteration history for the Turnbull algorithm
LRCL	requests likelihood ratio confidence intervals for distribution parameters
LRCLPER	requests likelihood ratio confidence intervals for distribution percentiles
LOCATION=*number* < LINIT >	specifies fixed or initial value of location parameter
MAXIT=*number*	specifies maximum number of iterations allowed for maximum likelihood fit
MAXITEM=*number1* <,*number2*>	*number1* specifies maximum number of iterations allowed for Turnbull algorithm. Iteration history will be printed in increments of *number2* if requested with ITPRINTEM. See "Interval-Censored Data" on page 1181 for details.
NOPCTILES	suppresses computation of percentiles for standard list of percentage points
NOPOLISH	suppress setting small interval probabilities to zero in Turnbull algorithm See "Interval-Censored Data" on page 1181 for details.

Table 37.26. Analysis Options (continued)

Option	Option Description
NPINTERVALS=	specifies type of nonparametric confidence interval displayed in a probability plot
POINTWISE | POINT	-pointwise confidence intervals for the CDF
| SIMULTANEOUS	
| SIMUL<*number1, number2*>	-simultaneous confidence intervals for the CDF. See "Simultaneous Confidence Intervals" on page 1187 for details.
PCTLIST=*number-list*	specifies list of percentages for which to compute percentile estimates. The *number-list* must be a list of numbers separated by blanks or commas. Each number in the list must be between 0 and 100.
PINTERVALS=	type of parametric pointwise confidence interval displayed in a probability plot. The default type is PROBABILITY, pointwise confidence intervals on cumulative failure probability.
LIKELIHOOD | LRCI	-likelihood ratio confidence intervals
| PERCENTILES | PER	-pointwise parametric confidence intervals for the percentiles of the fitted CDF
| PROBABILITY | CDF	-pointwise parametric confidence intervals for the cumulative failure probabilities
PPOS=	specifies plotting position type. See "Probability Plotting" beginning on page 1177 for details.
EXPRANK	-expected ranks
| MEDRANK	-median ranks
| MEDRANK1	-median ranks (exact formula)
| KM	-Kaplan-Meier
| MKM	-modified Kaplan-Meier (default)
|(NA | NELSONAALEN)	- Nelson-Aalen
PPOUT	request table of cumulative probabilities
PRINTPROBS	print intervals and associated probabilities for the Turnbull algorithm
PROBLIST=*number-list*	specifies list of initial values for Turnbull algorithm. See "Interval-Censored Data" on page 1181 for details.
PSTABLE=*number*	specifies stable parameterization. The *number* must be between zero and one. See "Stable Parameters" on page 1192 for further information.
READOUT	analyzes data with readout structure
SCALE=*number* < SCINIT >	specifies fixed or initial value of scale parameter
SHAPE=*number* < SHINIT >	specifies fixed or initial value of shape parameter
SINGULAR=*number*	specifies singularity criterion for matrix inversion
SURVTIME=*number-list*	requests survival function for values in *number-list*

Table 37.26. Analysis Options (continued)

Option	Option Description
THRESHOLD=*number*	specifies a fixed threshold parameter. See Table 37.40 for the distributions with a threshold parameter.
TOLLIKE=*number*	specifies criterion for convergence in the Turnbull algorithm. Default is 10^{-8}. See "Interval-Censored Data" on page 1181 for details.
TOLPROB=*number*	specifies criterion for setting interval probability to zero in the Turnbull algorithm. Default is 10^{-6}. See "Interval-Censored Data" on page 1181 for details.

Table 37.27. Probability Plot Layout Options

Option	Option Description
CENBIN	plots censored data as frequency counts rather than as individual points
CENSYMBOL= *symbol* | (*symbol list*)	specifies symbols for censored values. The *symbol* is one of the symbol names (plus, star, square, diamond, triangle, hash, paw, point, dot, circle) or a letter (A–Z). For overlaid plots for groups of data, you can specify different symbols for the groups with a list of symbols or letters, separated by blanks, enclosed in parentheses. If no CENSYMBOL option is specified, the symbol used for censored values is the same as for failures.
HOFFSET=*value*	specifies offset for horizontal axis
INBORDER	requests a border around probability plots
INTERTILE=*value*	specifies distance between tiles
LFIT=*linetype* | (*linetype list*)	line styles for fit lines and confidence curves in a probability plot. The *linetype list* is a list of numbers from 1 to 46 representing different linetypes, and can be separated by blanks or commas or can be a list in the form n_1 to n_2 <by n_3 >.
MISSING1	requests that missing values of first GROUP= variable be treated as a level of the variable
MISSING2	requests that missing values of second GROUP= variable be treated as a level of the variable
NCOLS=n	specifies number of columns plotted on a page
NOCENPLOT	suppresses plotting of censored data points
NOCONF	suppresses plotting of percentile confidence curves
NOFIT	suppresses plotting of fit line and percentile confidence curves
NOFRAME	suppresses frame around plotting area
NOINSET	suppresses inset
NOPPLEGEND	suppresses legend for overlaid probability plots

Table 37.27. Probability Plot Layout Options (continued)

Option	Option Description
NOPPOS	suppresses plotting of symbols for failures in a probability plot
NROWS=*n*	specifies number of rows plotted on a page
ORDER1=DATA | FORMATTED | FREQ | INTERNAL	specifies display order for values of the first GROUP= variable
ORDER2=DATA | FORMATTED | FREQ | INTERNAL	specifies display order for values of the second GROUP= variable
OVERLAY	requests overlaid plots for group variables
PCONFPLT	plots confidence intervals on probabilities for readout data
PPLEGEND = *legend-statement-name* | NONE	identifies LEGEND*n* statement to specify legend for overlaid probability plots
PPOSSYMBOL=*symbol* | (*symbol list*)	symbols representing failures on a probability plot
ROTATE	requests probability plots with probability scale on horizontal axis
SHOWMULTIPLES	display the count for multiple overlaying symbols
TURNVLABELS	vertically strings out characters in labels for vertical axis
VOFFSET=*value*	length of offset at upper end of vertical axis
WFIT=*linetype*	line width for fit line and confidence curves

Table 37.28. Reference Line Options

Option	Option Description
HREF < (INTERSECT) >=*value-list*	requests reference lines perpendicular to horizontal axis. If (INTERSECT) is specified, a second reference line perpendicular to the vertical axis is drawn that intersects the fit line at the same point as the horizontal axis reference line. If a horizontal axis reference line label is specified, the intersecting vertical axis reference line is labeled with the vertical axis value.
HREFLABELS=(*'label1' . . . 'labeln'*)	specifies labels for HREF= lines.

Table 37.28. Reference Line Options (continued)

Option	Option Description
HREFLABPOS=*n*	specifies vertical position of labels for HREF= lines. The valid values for *n* and the corresponding label placements are shown below. *n* | **label placement** 1 | top 2 | staggered from top 3 | bottom 4 | staggered from bottom 5 | alternating from top 6 | alternating from bottom
LHREF=*linetype*	specifies line style for HREF= lines
LVREF=*linetype*	specifies line style for VREF= lines
VREF < (INTERSECT) >=*value-list*	specifies reference lines perpendicular to vertical axis. If (INTERSECT) is specified, a second reference line perpendicular to the horizontal axis is drawn that intersects the fit line at the same point as the vertical axis reference line. If a vertical axis reference line label is specified, the intersecting horizontal axis reference line is labeled with the horizontal axis value.
VREFLABELS=*('label1' . . . 'labeln')*	specifies labels for VREF= lines
VREFLABPOS=*n*	specifies horizontal position of labels for VREF= lines. The valid values for *n* and the corresponding label placements are shown below. *n* | **label placement** 1 | left 2 | right

Table 37.29. Text Enhancement Options

Option	Option Description
FONT=*font*	specifies a software font for text
HEIGHT=*value*	specifies height of text used outside framed areas
INFONT=*font*	specifies a software font for text inside framed areas
INHEIGHT=*value*	specifies height of text inside framed areas

Table 37.30. Axis Options

Option	Option Description

LAXIS=*value1 to value2<by value3>*	specifies tick mark values for the lifetime axis. *value1*, *value2*, and *value3* must be numeric, and *value1* must be less than *value2*. The lower tick mark is *value1*. Tick marks are drawn at increments of *value3*. The last tick mark is the greatest value that does not exceed *value2*. If *value3* is omitted, a value of 1 is used. This method of specification of tick marks is not valid for logarithmic axes. Examples of LAXIS= lists are
	`laxis = -1 to 10`
	`laxis = 0 to 200 by 10`

LGRID=*number*	specifies a line style for all grid lines. The *number* is between 1 and 46 and specifies a linestyle for grids.
LIFELOWER | LLOWER=*number*	specifies the lower limit on the lifetime axis scale. The LLOWER option specifies *number* as the lower lifetime axis tick mark. The tick interval and the upper lifetime axis limit are determined automatically. This option has no effect if the LAXIS option is used.
LIFEUPPER | LUPPER=*number*	specifies the upper limit on the lifetime axis scale. The LUPPER option specifies *number* as the upper lifetime axis tick mark. The tick interval and the lower lifetime axis limit are determined automatically. This option has no effect if the LAXIS option is used.
MPGRID	adds a minor grid for the probability axis
MINORLOGGRID	adds a minor grid for log axes
NOGRID	suppresses grid lines
NOLLABEL	suppresses label for life, or analysis variable, axis
NOLTICK	suppresses tick marks and tick mark labels for lifetime or analysis variable axis
NOPLABEL	suppresses label for probability axis
NOPTICK	suppresses tick marks and tick mark labels for the probability axis
NTICK=*number*	specifies the number of tick intervals for the lifetime axis. This option has no effect if the LAXIS option is used.

Table 37.30. Axis Options (continued)

Option	Option Description
PCTLOWER | PLOWER=*number*	specifies the lower limit on probability axis scale
PCTUPPER | PUPPER=*number*	specifies the upper limit on probability axis scale
PAXISLABEL='*string*'	specifies a label for the probability axis
WAXIS=*n*	specifies the line thickness for axes and frame

Table 37.31. Color and Pattern Options

Option	Option Description
CAXIS=*color*	color for axis
CCENSOR=*color*	color for filling censor plot area
CENCOLOR=*color*	color for censor symbol
CFIT=*color* | (*color list*)	color for fit lines and confidence curves in a probability plot
CFRAME=*color*	color for frame
CFRAMESIDE=*color*	color for filling frame for row labels
CFRAMETOP=*color*	color for filling frame for column labels
CGRID=*color*	color for grid lines
CHREF=*color*	color for HREF= lines
CTEXT=*color*	color for text
CVREF=*color*	color for VREF= lines
PPOSCOLOR=*color* | (*color list*)	colors of symbols representing failures on a probability plot

Table 37.32. Graphics Catalog Options

Option	Option Description
DESCRIPTION='*string*'	description for graphics catalog member
NAME='*string*'	name for plot in graphics catalog

RELATIONPLOT Statement

<label:> **RELATIONPLOT** *variable<∗censor-variable(values)> <=group-variable>*
</options>;
<label:> **RELATIONPLOT** *(variable1 variable2) <=group-variables>*
</options>;

You use the RELATIONPLOT statement to create life-stress relation plots. A life-stress relation plot is a graphical tool for the analysis of data from accelerated life tests. The plot is a display of the relationship between life and *stress*, such as temperature or voltage. You can also use the RELATIONPLOT statement to display a

probability plot alongside the relation plot. See Figure 37.8 on page 1094 for an example of a relation plot.

You can specify the keyword RPLOT as an alias for RELATIONPLOT. You can use any number of RELATIONPLOT statements after a PROC RELIABILITY statement. You can specify an optional *label* to distinguish between multiple RELATIONPLOT statements in the output.

See "Analysis of Accelerated Life Test Data" on page 1090 for an example using the RELATIONPLOT statement.

To create a life-stress relation plot, you must specify one *variable*. If your data are right censored, you must specify a *censor-variable* and, in parentheses, the *values* of the *censor-variable* that correspond to censored data values. You must specify one *group-variable* to represent the values of stress. The *group-variable* must be a numeric variable.

The RELATIONPLOT statement plots the uncensored values of your data given by *variable* versus the values of the *group-variable*. You can optionally display a box-plot of the values of the data. You can also plot percentiles of the distribution fitted to the data. The RELATIONPLOT statement produces the same tabular output as the PROBPLOT statement, and all the analysis options are the same as for the PROBPLOT statement.

The elements of the RELATIONPLOT statement are described as follows.

variable
represents the data for which a plot is to be produced. The *variable* must be a numeric variable in the input data set.

censor-variable(values)
indicates which observations in the input data set are right censored. You specify the values of *censor-variable* that represent censored observations by placing those values in parentheses after the variable name. If your data are not right censored, then you omit the specification of *censor-variable*; otherwise, *censor-variable* must be a numeric variable in the input data set.

(variable1 variable2)
is another method of specifying the data for which a life-stress plot is to be produced. You can use this syntax in a situation where uncensored, interval-censored, left-censored and right-censored values occur in the same set of data. Table 37.23 shows how you use this syntax to specify different types of censoring by using combinations of missing and nonmissing values. See "Lognormal Analysis with Arbitrary Censoring" on page 1100 for an example of using this syntax to create a probability plot.

group-variable
is a group variable. The *group-variable* must be a numeric variable in the input data set.

options
control the features of the relation plot. All *options* are specified after the slash (/) in

the RELATIONPLOT statement. The "Summary of Options" section, which follows, lists all options by function.

The only type of relation plot currently available for interval data is the type in which percentiles of the fitted distribution are plotted at each stress level.

Summary of Options

Table 37.33. Analysis Options

Option	Option Description
CONFIDENCE=*number*	specifies the confidence coefficient for all confidence intervals. The *number* must be between 0 and 1. The default value is 0.95
CONVERGE=*number*	specifies the convergence criterion for maximum likelihood fit. See "Maximum Likelihood Estimation" on page 1188 for details.
CONVH=*number*	specifies the convergence criterion for the relative Hessian convergence criterion See "Maximum Likelihood Estimation" on page 1188 for details.
CORRB	requests parameter correlation matrix
COVB	requests parameter covariance matrix
FITTYPE=	specifies method of estimating distribution parameters
LSYX	-least squares fit to the probability plot. The probability axis is the dependent variable.
| LSXY	-least squares fit to the probability plot. The lifetime axis is the dependent variable.
| MLE	-maximum likelihood (default)
| MODEL	-use the fit from the preceding MODEL statement
| NONE	-no fit is computed
| REGRESSION	-use the fit from the preceding MODEL statement. Non-linear relations and percentiles from models using two independent variables can be plotted.
| WEIBAYES <(CONFIDENCE|CONF=*number*)>	-Weibayes method The *number* is the confidence coefficient for the Weibayes fit. The *number* is between 0 and 1, with a default value of 0.95.
ITPRINT	requests iteration history for maximum likelihood fit
LRCL	requests likelihood ratio confidence intervals for distribution parameters
LRCLPER	requests likelihood ratio confidence intervals for distribution percentiles
LOCATION=*number* < LINIT >	specifies fixed or initial value of location parameter
MAXIT=*number*	specifies maximum number of iterations allowed for maximum likelihood fit

Table 37.33. Analysis Options (continued)

Option	Option Description
NOPCTILES	suppress computation of percentiles
PCTLIST=*number-list*	specifies list of percentages for which to compute percentile estimates. The *number-list* must be a list of numbers separated by blanks or commas. Each number in the list must be between 0 and 100.
PPOS=	specifies plotting position type. See "Probability Plotting" beginning on page 1177 for details.
EXPRANK	-expected ranks
| MEDRANK	-median ranks
| MEDRANK1	-median ranks (exact formula)
| KM	-Kaplan-Meier
| MKM	-modified Kaplan-Meier (default)
| (NA | NELSONAALEN)	- Nelson-Aalen
PPOUT	request table of cumulative probabilities
PSTABLE=*number*	specifies stable parameterization. The *number* must be between zero and one. See "Stable Parameters" on page 1192 for further information.
RELATION=ARRHENIUS | ARRHENIUS2 | POWER | LINEAR | LOGISTIC	specifies type of relationship between life and stress. This determines the horizontal scale used in the relation plot. See Table 37.49 on page 1191 for definitions of the transformations.
READOUT	analyzes data with readout structure
SCALE=*number* < SCINIT >	specifies fixed or initial value of scale parameter
SHAPE=*number* < SHINIT >	specifies fixed or initial value of shape parameter
SINGULAR=*number*	specifies singularity criterion for matrix inversion
SURVTIME=*number-list*	requests that survival function be computed for values in *number-list*
THRESHOLD=*number*	specifies a fixed threshold parameter. See Table 37.40 for the distributions with a threshold parameter.
variable=*number list*	allows plots of percentiles from a regression model when two independent variables are used in a MODEL statement *effect list*. The FIT=REGRESSION option must be used with this option. Percentile plots are created for each value of the independent *variable* in the *number list*. *number list* is a list of numeric values separated by blanks or commas, or in the form of a list n_1 to n_2 <by n_3 >.

Table 37.34. Plot Layout Options

Option	Option Description
CENSYMBOL=*symbol* | (*symbol list*)	specifies symbols for censored values. The *symbol* is one of the symbol names (plus, star, square, diamond, triangle, hash, paw, point, dot, circle) or a letter (A–Z). If you are creating overlaid plots for groups of data, you can specify different symbols for the groups with a list of symbols or letters, separated by blanks, enclosed in parentheses. If no CENSYMBOL option is specified, the symbol used for censored values is the same as for failures.
HOFFSET=*value*	specifies an offset for horizontal axis
INBORDER	requests a border around plots
LBOXES=*number*	specifies a line style for boxplots
LFIT=*linetype* | (*linetype list*))	line styles for fit lines and confidence curves in a probability plot. The *linetype list* is a list of numbers from 1 to 46 representing different linetypes, and can be separated by blanks or commas or can be a list in the form n_1 to n_2 <by n_3 >.
LPLOTFIT=*linetype* | (*linetype list*))	line styles for percentile lines. *linetype list* is a list of numbers representing different linetypes, and can be separated by blanks or commas or can be a list in the form n_1 to n_2 <by n_3 >.
NOCENPLOT	suppresses plotting of censored data points
NOCONF	suppresses plotting of percentile confidence curves
NOFIT	suppresses plotting of fit line and percentile confidence curves
NOFRAME	suppresses frame around plotting area
NOPPLEGEND	suppresses legend for overlaid probability plots
NOPPOS	suppresses plotting of symbols for failures in a probability plot
NORPLEGEND	suppresses legend for relation plot
PINTERVALS=	type of parametric pointwise confidence interval displayed in a probability plot. The default type is PROBABILITY, pointwise confidence intervals on cumulative failure probability.
LIKELIHOOD | LRCI	-likelihood ratio confidence intervals
| PERCENTILES | PER	-pointwise parametric confidence intervals for the percentiles of the fitted CDF

Table 37.34. Plot Layout Options (continued)

Option	Option Description
| PROBABILITY | CDF	-pointwise parametric confidence intervals for the cumulative failure probabilities
PLOTDATA <DATA | MEDIANS | BOXES>	requests that the data be plotted on the relationplot and specifies the representation of the data populations to be plotted
PLOTFIT <*number-list*>	specifies that percentiles of the fitted distribution be plotted on the relation plot. The optional *number-list* is a list of percentiles (between 0 and 100), and, if not specified, the 50th percentile (median) is plotted.
PPLEGEND = *legend-statement-name* | NONE	identifies a LEGEND*n* statement to specify legend for overlaid probability plots
PPLOT	places a probability plot on the same page as the relation plot
RCENSYMBOL=*symbol* | (*symbol list*)	symbols representing right censored and left censored observations in a relation plot. The *symbol* is one of the symbol names (plus, star, square, diamond, triangle, hash, paw, point, dot, circle) or a letter (A–Z).
RPLEGEND = *legend-statement-name* | NONE	identifies a LEGEND*n* statement to specify legend for the relation plot
SHOWMULTIPLES	display the count for multiple overlaying symbols
TURNVLABELS	vertically strings out characters in labels for vertical axis
VOFFSET=*value*	specifies length of offset at upper end of vertical axis
WFIT=*linetype*	specifies line width for fit line and confidence curves

Table 37.35. Reference Line Options

Option	Option Description
HREF < (INTERSECT) >=*value-list*	requests reference lines perpendicular to horizontal axis. If (INTERSECT) is specified, a second reference line perpendicular to the vertical axis is drawn that intersects the fit line at the same point as the horizontal axis reference line. If a horizontal axis reference line label is specified, the intersecting vertical axis reference line is labeled with the vertical axis value.
HREFLABELS=(*'label1' . . . 'labeln'*)	specifies labels for HREF= lines

Table 37.35. Reference Line Options (continued)

Option	Option Description
HREFLABPOS=*n*	specifies vertical position of labels for HREF= lines. The valid values for *n* and the corresponding label placements are shown below. *n* | **label placement** 1 | top 2 | staggered from top 3 | bottom 4 | staggered from bottom 5 | alternating from top 6 | alternating from bottom
LHREF=*linetype*	specifies a line style for HREF= lines
LSREF=*linetype*	specifies a line style for SREF= lines
LVREF=*linetype*	specifies a line style for VREF= lines
SREF=*value-list*	specifies reference lines perpendicular to horizontal stress axis
SREFLABELS=*('label1' . . . 'labeln')*	specifies labels for SREF= lines
SREFLABPOS=*n*	specifies horizontal position of labels for SREF= lines. The valid values for *n* and the corresponding label placements are shown below. *n* | **label placement** 1 | top 2 | staggered from top 3 | bottom 4 | staggered from bottom
VREF < (INTERSECT) >=*value-list*	requests reference lines perpendicular to vertical axis.If (INTERSECT) is specified, a second reference line perpendicular to the horizontal axis is drawn that intersects the fit line at the same point as the vertical axis reference line. If a vertical axis reference line label is specified, the intersecting horizontal axis reference line is labeled with the horizontal axis value.
VREFLABELS=*('label1' . . . 'labeln')*	specifies labels for VREF= lines

Table 37.35. Reference Line Options (continued)

Option	Option Description
VREFLABPOS=*n*	specifies horizontal position of labels for VREF= lines. The valid values for *n* and the corresponding label placements are shown below. *n* | **label placement** 1 | left 2 | right

Table 37.36. Text Enhancement Options

Option	Option Description
FONT=*font*	specifies a software font for text
HEIGHT=*value*	specifies height of text used outside framed areas
INFONT=*font*	specifies a software font for text inside framed areas
INHEIGHT=*value*	specifies height of text inside framed areas

Table 37.37. Axis Options

Option	Option Description

LAXIS=*value1 to value2<by value3>*	specifies tick mark values for the lifetime axis. *value1*, *value2*, and *value3* must be numeric, and *value1* must be less than *value2*. The lower tick mark is *value1*. Tick marks are drawn at increments of *value3*. The last tick mark is the greatest value that does not exceed *value2*. If *value3* is omitted, a value of 1 is used. This method of specification of tick marks is not valid for logarithmic axes. Examples of LAXIS= lists are
	`laxis = -1 to 10`
	`laxis = 0 to 200 by 10`

LGRID=*number*	specifies a line style for all grid lines. The *number* is between 1 and 46 and specifies a linestyle for grids.

Table 37.37. Axis Options (continued)

Option	Option Description
LIFELOWER | LLOWER=*number*	specifies the lower limit on the lifetime axis scale. The LLOWER option specifies *number* as the lower lifetime axis tick mark. The tick interval and the upper lifetime axis limit are determined automatically. This option has no effect if the LAXIS option is used.
LIFEUPPER | LUPPER=*number*	specifies the upper limit on the lifetime axis scale. The LUPPER option specifies *number* as the upper lifetime axis tick mark. The tick interval and the lower lifetime axis limit are determined automatically. This option has no effect if the LAXIS option is used.
MPGRID	adds a minor grid for the probability axis
MINORLOGGRID	adds a minor grid for log axes
NOGRID	suppresses grid lines
NOLLABEL	suppresses label for life, or analysis variable, axis
NOLTICK	suppresses tick marks and tick mark labels for lifetime or analysis variable axis
NOPLABEL	suppresses label for probability axis
NOPTICK	suppresses tick marks and tick mark labels for probability axis
NOSLABEL	suppresses label for stress axis
NOSTICK	suppresses tick marks and tick mark labels for stress axis
NSTRESSTICK=*number*	specifies the number of tick intervals for stress axis for relation plot
NTICK=*number*	specifies the number of tick intervals for the lifetime axis. This option has no effect if the LAXIS option is used.
PCTLOWER | PLOWER=*number*	specifies lower limit on probability axis scale
PCTUPPER | PUPPER=*number*	specifies upper limit on probability axis scale
STRESSLOWER | SLOWER=*number*	specifies lower limit on stress axis scale
STRESSUPPER | SUPPER=*number*	specifies upper limit on stress axis scale
PAXISLABEL='*string*'	specifies label for probability axis
WAXIS=n	specifies line thickness for axes and frame

Table 37.38. Graphics Catalog Options

Option	Option Description
DESCRIPTION='*string*'	description for graphics catalog member
NAME='*string*'	name for plot in graphics catalog

Table 37.39. Color and Pattern Options

Option	Option Description
CAXIS=*color*	color for axis
CBOXES=*color*	color for box frame for boxplots
CBOXFILL=*color*	color for filling boxes for boxplots
CCENSOR=*color*	color for filling censor plot area
CENCOLOR=*color*	color for censor symbol
CFIT=*color* | (*color list*)	color for fit lines and confidence curves in a probability plot
CFRAME=*color*	color for frame
CGRID=*color*	color for grid lines
CHREF=*color*	color for HREF= lines
CPLOTFIT=*color* | (*color list*)	colors for percentile lines
CSREF=*color*	color for SREF= lines
CTEXT=*color*	color for text
CVREF=*color*	color for VREF= lines
RCENCOLOR=*color* | (*color list*)	colors for the symbols representing uncensored, right censored, and left censored observations in a relation plot

UNITID Statement

UNITID *variable*;

The UNITID statement names a *variable* in the input data set that is used to identify each individual unit in an MCFPLOT statement. The value of the UNITID variable for an observation corresponds to the name of the unit in the study for which a repair or end of history has occurred. See "Analysis of Recurrence Data on Repairs" on page 1118 for an example using the UNITID statement with the MCFPLOT statement.

Details

Abbreviations and Notation

The following abbreviations and notation are used in this section.

CDF	cumulative distribution function: $F(x) = Pr\{X \leq x\}$
log	base e logarithm
$\log_{10}$	base 10 logarithm
Reliability or Survivor function	$R(x) = Pr\{X > x\}$
x_p	$p \times 100\%$ percentile: $Pr\{X \leq x_p\} = p$

Types of Lifetime Data

This section describes various types of data that you can analyze with the RELIABILITY procedure.

Lifetime data for which the values of all sample units are observed are called *complete* data. This means that the failure times are observed for all units.

Many practical problems in life data analysis involve data for which some units are unfailed. The failure time for an unfailed unit is known only to be greater than the last running time. This type of data is said to be *right censored*, and the censoring time is used in the analysis of the data. Data for which censoring times are intermixed with failure times are sometimes called *multiply censored* or *progressively censored*.

Failure times may be known only to be less than some value. This type of data is called *left censored*.

Another common situation is where the failure times of units are not known exactly, but time intervals that contain the failure times are known. This type of data is called *interval censored*.

Interval-censored data for which all units share common interval endpoints are called *readout*, *inspection*, or *grouped* data.

Arbitrarily censored data can contain a combination of failures, right-, left-, and interval-censored data.

Probability Distributions

This section describes the probability distributions available in the RELIABILITY procedure for probability plotting and parameter estimation.

PROBPLOT and RELATIONPLOT Statements

Probability plots can be constructed for each of the probability distributions in Table 37.40. Estimates of two distribution parameters (*location* and *scale* or *scale* and *shape*) are computed by maximum likelihood or by least squares fitted to points on

the probability plot. If one of the parameters is specified as fixed, the other is estimated. In addition, you can specify a fixed *threshold*, or *shift*, parameter for those distributions for which a threshold parameter is indicated in Table 37.40. If you do not specify a threshold parameter, the threshold is set to 0.

Note that you should not interpret the parameters μ and σ as representing the means and standard deviations for all of the distributions in Table 37.40. The normal is the only distribution in Table 37.40 for which this is the case.

Table 37.40. Distributions and Parameters for PROBPLOT and RELATIONPLOT Statements

Distribution	Density Function	Parameters			
		Location	Scale	Shape	Threshold
Normal	$\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$	μ	σ		
Lognormal	$\frac{1}{\sqrt{2\pi}\sigma(x-\theta)}\exp\left(-\frac{(\log(x-\theta)-\mu)^2}{2\sigma^2}\right)$	μ	σ		θ
Lognormal (base 10)	$\frac{\log(10)}{\sqrt{2\pi}\sigma(x-\theta)}\exp\left(-\frac{(\log_{10}(x-\theta)-\mu)^2}{2\sigma^2}\right)$	μ	σ		θ
Extreme Value	$\frac{1}{\sigma}\exp\left(\frac{x-\mu}{\sigma}\right)\exp\left(-\exp\left(\frac{x-\mu}{\sigma}\right)\right)$	μ	σ		
Weibull	$\frac{\beta}{\alpha^\beta}(x-\theta)^{\beta-1}\exp\left(-\left(\frac{x-\theta}{\alpha}\right)^\beta\right)$		α	β	θ
Exponential	$\frac{1}{\alpha}\exp\left(-\left(\frac{x-\theta}{\alpha}\right)\right)$		α		θ
Logistic	$\frac{\exp\left(\frac{x-\mu}{\sigma}\right)}{\sigma\left[1+\exp\left(\frac{x-\mu}{\sigma}\right)\right]^2}$	μ	σ		
Log-logistic	$\frac{\exp\left(\frac{\log(x-\theta)-\mu}{\sigma}\right)}{(x-\theta)\sigma\left[1+\exp\left(\frac{\log(x-\theta)-\mu}{\sigma}\right)\right]^2}$	μ	σ		θ

The exponential distribution shown in Table 37.40 is a special case of the Weibull distribution with $\beta = 1$. The remaining distributions in Table 37.40 are related to one another as shown in Table 37.41. The threshold parameter, θ, is assumed to be 0 in Table 37.41.

Table 37.41. Relationship among Life Distributions

Distribution of T	Parameters		Distribution of Y=logT	Parameters	
Lognormal	μ	σ	Normal	μ	σ
Weibull	α	β	Extreme Value	$\mu = \log\alpha$	$\sigma = \frac{1}{\beta}$
Log-logistic	μ	σ	Logistic	μ	σ

MODEL Statement

All of the distributions in Table 37.40 are available for regression model estimation using the MODEL statement. In addition, the generalized gamma distribution with the following probability density function $f(t)$ is available for regression model estimation in the MODEL statement.

$$f(t) = \frac{|\lambda|}{t\sigma\Gamma(\lambda^{-2})}(\lambda^{-2})^{(\lambda^{-2})} \exp\left[\lambda^{-2}\left(\lambda\left(\frac{\log(t)-\mu}{\sigma}\right) - \exp\left(\lambda\left(\frac{\log(t)-\mu}{\sigma}\right)\right)\right)\right]$$

If a lifetime T has the generalized gamma distribution, then the logarithm of the lifetime $X = \log(T)$ has the generalized log-gamma distribution, with the following probability density function $g(x)$. When the gamma distribution is specified, the logarithms of the lifetimes are used as responses, and the generalized log-gamma distribution is used to estimate the parameters by maximum likelihood.

$$g(x) = \frac{|\lambda|}{\sigma\Gamma(\lambda^{-2})}(\lambda^{-2})^{(\lambda^{-2})} \exp\left[\lambda^{-2}\left(\lambda\left(\frac{x-\mu}{\sigma}\right) - \exp\left(\lambda\left(\frac{x-\mu}{\sigma}\right)\right)\right)\right]$$

Refer to Lawless (1982, p. 240) and Meeker and Escobar (1998, p. 101) for a description of the generalized gamma and generalized log-gamma distributions.

When $\lambda = 1$, the generalized log-gamma distribution reduces to the extreme value distribution with parameters μ and σ. In this case, the log lifetimes have the extreme value distribution, or, equivalently, the lifetimes have the Weibull distribution with parameters $\alpha = \exp(\mu)$ and $\beta = 1/\sigma$. When $\lambda = 0$, the generalized log-gamma reduces to the normal distribution with parameters μ and σ. In this case, the (unlogged) lifetimes have the lognormal distribution with parameters μ and σ. This chapter uses the notation μ for the *location*, σ for the *scale*, and λ for the *shape* parameters for the generalized log-gamma distribution.

ANALYZE Statement

You can use the ANALYZE statement to compute parameter estimates and other statistics for the distributions in Table 37.40. In addition, you can compute estimates for the binomial and Poisson distributions. The forms of these distributions are shown in Table 37.42.

Table 37.42. Binomial and Poisson Distributions

Distribution	Pr{Y=y}	Parameter	Parameter Name
Binomial	$\binom{n}{y} p^y(1-p)^{n-y}$	p	binomial probability
Poisson	$\frac{\mu^y}{y!}\exp(-\mu)$	μ	Poisson mean

Probability Plotting

Probability plots are useful tools for the display and analysis of lifetime data. Refer to Abernethy (1996) for examples using probability plots in the analysis of reliability data. Probability plots use a special scale so that a cumulative distribution function (CDF) plots as a straight line. Thus, if lifetime data are a sample from a distribution, the CDF estimated from the data plots approximately as a straight line on a probability plot for the distribution.

You can use the RELIABILITY procedure to construct probability plots for data that are complete, right censored, or interval censored (in readout form) for each of the probability distributions in Table 37.40.

A random variable Y belongs to a *location-scale* family of distributions if its CDF F is of the form

$$Pr\{Y \leq y\} = F(y) = G\left(\frac{y-\mu}{\sigma}\right)$$

where μ is the location parameter, and σ is the scale parameter. Here, G is a CDF that cannot depend on any unknown parameters, and G is the CDF of Y if $\mu = 0$ and $\sigma = 1$. For example, if Y is a normal random variable with mean μ and standard deviation σ,

$$G(u) = \Phi(u) = \int_{-\infty}^{u} \frac{1}{\sqrt{2\pi}} \exp(-\frac{u^2}{2})\, du$$

and

$$F(y) = \Phi\left(\frac{y-\mu}{\sigma}\right)$$

Of the distributions in Table 37.40, the normal, extreme value, and logistic distributions are location-scale models. As shown in Table 37.41, if T has a lognormal, Weibull, or log-logistic distribution, then $\log(T)$ has a distribution that is a location-scale model. Probability plots are constructed for lognormal, Weibull, and log-logistic distributions by using $\log(T)$ instead of T in the plots.

Let $y_{(1)} \leq y_{(2)} \leq \ldots \leq y_{(n)}$ be ordered observations of a random sample with distribution function $F(y)$. A probability plot is a plot of the points $y_{(i)}$ against $m_i = G^{-1}(a_i)$, where $a_i = \hat{F}(y_i)$ is an estimate of the CDF $F(y_{(i)}) = G\left(\frac{y_{(i)}-\mu}{\sigma}\right)$. The points a_i are called *plotting positions*. The axis on which the points m_is are plotted is usually labeled with a probability scale (the scale of a_i).

If F is one of the location-scale distributions, then y is the lifetime; otherwise, the log of the lifetime is used to transform the distribution to a location-scale model.

If the data actually have the stated distribution, then $\hat{F} \approx F$,

$$m_i = G^{-1}(\hat{F}(y_i) \approx G^{-1}(G\left(\frac{y_{(i)}-\mu}{\sigma}\right)) = \frac{y_{(i)}-\mu}{\sigma}$$

and points $(y_{(i)}, m_i)$ should fall approximately on a straight line.

There are several ways to compute plotting positions from failure data. These are discussed in the next two sections.

Complete and Right-Censored Data

The censoring times must be taken into account when you compute plotting positions for right-censored data. The RELIABILITY procedure provides several methods for computing plotting positions. These are specified with the PPOS= option in the ANALYZE, PROBPLOT, and RELATIONPLOT statements. All of the methods give similar results, as illustrated in the following sections, "Expected Ranks, Kaplan-Meier, and Modified Kaplan-Meier Methods" and "Median Ranks."

Expected Ranks, Kaplan-Meier, and Modified Kaplan-Meier Methods

Let $y_{(1)} \leq y_{(2)} \leq \ldots \leq y_{(n)}$ be ordered observations of a random sample including failure times and censor times. Order the data in increasing order. Label all the data with reverse ranks r_i, with $r_1 = n, \ldots, r_n = 1$. For the failure corresponding to reverse rank r_i, compute the reliability, or survivor function estimate

$$R_i = \left[\frac{r_i}{r_i + 1}\right] R_{i-1}$$

with $R_0 = 1$. The expected rank plotting position is computed as $a_i = 1 - R_i$. The option PPOS=EXPRANK specifies the expected rank plotting position.

For the Kaplan-Meier method,

$$R_i = \left[\frac{r_i - 1}{r_i}\right] R_{i-1}$$

The Kaplan-Meier plotting position is then computed as $a'_i = 1 - R_i$. The option PPOS=KM specifies the Kaplan-Meier plotting position.

For the modified Kaplan-Meier method, use

$$R'_i = \frac{R_i + R_{i-1}}{2}$$

where R_i is computed from the Kaplan-Meier formula with $R_0 = 1$. The plotting position is then computed as $a''_i = 1 - R'_i$. The option PPOS=MKM specifies the modified Kaplan-Meier plotting position. If the PPOS option is not specified, the modified Kaplan-Meier plotting position is used as the default method.

For complete samples, $a_i = i/(n+1)$ for the expected rank method, $a'_i = i/n$ for the Kaplan-Meier method, and $a''_i = (i - .5)/n$ for the modified Kaplan-Meier method. If the largest observation is a failure for the Kaplan-Meier estimator, then $F_n = 1$ and the point is not plotted. These three methods are shown for the field winding data in Table 37.43 and Table 37.44.

Table 37.43. Expected Rank Plotting Position Calculations

Ordered Observation	Reverse Rank	$r_i/(r_i+1)$	$\times R_{i-1}$	$= R_i$	$a_i = 1 - R_i$
31.7	16	16/17	1.0000	0.9411	0.0588
39.2	15	15/16	0.9411	0.8824	0.1176
57.5	14	14/15	0.8824	0.8235	0.1765
65.0+	13				
65.8	12	12/13	0.8235	0.7602	0.2398
70.0	11	11/12	0.7602	0.6968	0.3032
75.0+	10				
75.0+	9				
87.5+	8				
88.3+	7				
94.2+	6				
101.7+	5				
105.8	4	4/5	0.6968	0.5575	0.4425
109.2+	3				
110.0	2	2/3	0.5575	0.3716	0.6284
130.0+	1				

+ Censored Times

Table 37.44. Kaplan-Meier and Modified Kaplan-Meier Plotting Position Calculations

Ordered Observation	Reverse Rank	$(r_i-1)/r_i$	$\times R_{i-1}$	$= R_i$	$a_i' = 1 - R_i$	a_i''
31.7	16	15/16	1.0000	0.9375	0.0625	0.0313
39.2	15	14/15	0.9375	0.8750	0.1250	0.0938
57.5	14	13/14	0.8750	0.8125	0.1875	0.1563
65.0+	13					
65.8	12	11/12	0.8125	0.7448	0.2552	0.2214
70.0	11	10/11	0.7448	0.6771	0.3229	0.2891
75.0+	10					
75.0+	9					
87.5+	8					
88.3+	7					
94.2+	6					
101.7+	5					
105.8	4	3/4	0.6771	0.5078	0.4922	0.4076
109.2+	3					
110.0	2	1/2	0.5078	0.2539	0.7461	0.6192
130.0+	1					

+ Censored Times

Nelson-Aalen

Estimate the cumulative hazard function by

$$H_i = \frac{1}{r_i} + H_{i-1}$$

with $H_0 = 0$. The reliability is $R_i = \exp(-H_i)$, and the plotting position, or CDF is $a_i''' = 1 - R_i$. You can show that $R_{KM} < R_{NA}$ for all ages. The Nelson-Aalen method is shown for the field winding data in Table 37.45.

Table 37.45. Nelson-Aalen Plotting Position Calculations

Ordered Observation	Reverse Rank	$1/r_i$	$+H_{i-1}$	$= H_i$	$a_i''' = 1 - \exp(-H_i)$
31.7	16	1/16	0.0000	0.0625	0.0606
39.2	15	1/15	0.0625	0.1292	0.1212
57.5	14	1/14	0.1292	0.2006	0.1818
65.0+	13				
65.8	12	1/12	0.2006	0.2839	0.2472
70.0	11	1/11	0.2839	0.3748	0.3126
75.0+	10				
75.0+	9				
87.5+	8				
88.3+	7				
94.2+	6				
101.7+	5				
105.8	4	1/4	0.3748	0.6248	0.4647
109.2+	3				
110.0	2	1/2	0.6248	1.1248	0.6753
130.0+	1				

+ Censored Times

Median Ranks

Refer to Abernethy (1996) for a discussion of the methods described in this section. Let $y_{(1)} \leq y_{(2)} \leq \ldots \leq y_{(n)}$ be ordered observations of a random sample including failure times and censor times. A failure order number j_i is assigned to the *i*th failure: $j_i = j_{i-1} + \Delta$, where $j_0 = 0$. The increment Δ is initially 1 and is modified when a censoring time is encountered in the ordered sample. The new increment is computed as

$$\Delta = \frac{(n+1) - \text{ previous failure order number}}{1 + \text{ number of items beyond previous censored item}}$$

The plotting position is computed for the ith failure time as

$$a_i = \frac{j_i - .3}{n + .4}$$

For complete samples, the failure order number j_i is equal to i, the order of the failure in the sample. In this case, the preceding equation for a_i is an approximation to the median plotting position computed as the median of the ith-order statistic from the uniform distribution on (0, 1). In the censored case, j_i is not necessarily an integer, but the preceding equation still provides an approximation to the median plotting position. The PPOS=MEDRANK option specifies the median rank plotting position.

For complete data, an alternative method of computing the median rank plotting position for failure i is to compute the exact median of the distribution of the ith order statistic of a sample of size n from the uniform distribution on (0,1). If the data are right censored, the adjusted rank j_i, as defined in the preceding paragraph, is used in place of i in the computation of the median rank. The PPOS=MEDRANK1 option specifies this type of plotting position.

Nelson (1982, p.148) provides the following example of multiply right-censored failure data for field windings in electrical generators. Table 37.46 shows the data, the intermediate calculations, and the plotting positions calculated by exact (a'_i) and approximate (a_i) median ranks.

Table 37.46. Median Rank Plotting Position Calculations

Ordered Observation	Increment Δ	Failure Order Number j_i	a_i	a'_i
31.7	1.0000	1.0000	0.04268	0.04240
39.2	1.0000	2.0000	0.1037	0.1027
57.5	1.0000	3.0000	0.1646	0.1637
65.0+	1.0769			
65.8	1.0769	4.0769	0.2303	0.2294
70.0	1.0769	5.1538	0.2960	0.2953
75.0+	1.1846			
75.0+	1.3162			
87.5+	1.4808			
88.3+	1.6923			
94.2+	1.9744			
101.7+	2.3692			
105.8	2.3692	7.5231	0.4404	0.4402
109.2+	3.1590			
110.0	3.1590	10.6821	0.6331	0.6335
130.0+	6.3179			

+ Censored Times

Interval-Censored Data

Readout Data

The RELIABILITY procedure can create probability plots for interval-censored data when all units share common interval endpoints. This type of data is called *readout* data in the RELIABILITY procedure. Estimates of the cumulative distribution function are computed at times corresponding to the interval endpoints. Right censoring can also be accommodated if the censor times correspond to interval endpoints. See "Weibull Analysis of Interval Data with Common Inspection Schedule" on page 1096 for an example of a Weibull plot and analysis for interval data.

Table 37.47 illustrates the computational scheme used to compute the CDF estimates. The data are failure data for microprocessors (Nelson 1990, p.147). In Table 37.47, t_i are the interval upper endpoints, in hours, f_i is the number of units failing in interval i, and n_i is the number of unfailed units at the beginning of interval i.

Note that there is right censoring as well as interval censoring in these data. For example, two units fail in the interval (24, 48) hours, and there are 1414 unfailed units at the beginning of the interval, 24 hours. At the beginning of the next interval, (48, 168) hours, there are 573 unfailed units. The number of unfailed units that are removed from the test at 48 hours is $1414 - 2 - 573 = 839$ units. These are right-censored units.

The reliability at the end of interval i is computed recursively as

$$R_i = (1 - (f_i/n_i))R_{i-1}$$

with $R_0 = 1$. The plotting position is $a_i = 1 - R_i$.

Table 37.47. Interval-Censored Plotting Position Calculations

Interval i	Interval Endpoint t_i	f_i/n_i	$R_i^{'} = 1-(f_i/ni)$	$R_i = R_i^{'}R_{i-1}$	$a_i = 1 - R_i$
1	6	6/1423	0.99578	0.99578	.00421
2	12	2/1417	0.99859	0.99438	.00562
3	24	0/1415	1.00000	0.99438	.00562
4	48	2/1414	0.99859	0.99297	.00703
5	168	1/573	0.99825	0.99124	.00876
6	500	1/422	0.99763	0.98889	.01111
7	1000	2/272	0.99265	0.98162	.01838
8	2000	1/123	0.99187	0.97364	.02636

Arbitrarily Censored Data

The RELIABILITY procedure can create probability plots for data that consists of combinations of exact, left-censored, right-censored, and interval-censored lifetimes. Unlike the method in the previous section, failure intervals need not share common endpoints, although if the intervals share common endpoints, the two methods give the same results. The RELIABILITY procedure uses an iterative algorithm developed by Turnbull (1976) to compute a nonparametric maximum likelihood estimate of the cumulative distribution function for the data. Since the technique is maximum likelihood, standard errors of the cumulative probability estimates are computed from the inverse of the associated Fisher information matrix. A technique developed by Gentleman and Geyer (1994) is used to check for convergence to the maximum likelihood estimate. Also see Meeker and Escobar (1998, chap. 3) for more information.

Although this method applies to more general situations, where the intervals may be overlapping, the example of the previous section will be used to illustrate the method. Table 37.48 contains the microprocessor data of the previous section, arranged in intervals. A missing (.) lower endpoint indicates left censoring, and a missing upper endpoint indicates right censoring. These can be thought of as semi-infinite intervals with lower (upper) endpoint of negative (positive) infinity for left (right) censoring.

Table 37.48. Interval-Censored Data

Lower Endpoint	Upper Endpoint	Number Failed
.	6	6
6	12	2
24	48	2
24	.	1
48	168	1
48	.	839
168	500	1
168	.	150
500	1000	2
500	.	149
1000	2000	1
1000	.	147
2000	.	122

The following SAS program will compute the Turnbull estimate and create a lognormal probability plot.

```
   data micro;
      input t1 t2 f ;
      cards;
      . 6 6
      6 12 2
      12 24 0
      24 48 2
      24 .  1
      48 168 1
      48 .   839
      168 500 1
      168 .   150
      500 1000 2
      500 .    149
      1000 2000 1
      1000 . 147
      2000 . 122
      ;
   run;

proc reliability data=micro;
   distribution lognormal;
   freq f;
   pplot ( t1 t2 ) / itprintem
                     printprobs
                     maxitem = ( 1000, 25 )
                     nofit
                     npintervals = simul
                     ppout;
run;
```

The nonparametric maximum likelihood estimate of the CDF can only increase on certain intervals, and must remain constant between the intervals. The Turnbull algorithm first computes the intervals on which the nonparametric maximum likelihood estimate of the CDF can increase. The algorithm then iteratively estimates the probability associated with each interval. The ITPRINTEM option along with the PRINTPROBS option instructs the procedure to print the intervals on which probability increases can occur and the iterative history of the estimates of the interval probabilities. The PPOUT option requests tabular output of the estimated CDF, standard errors, and confidence limits for each cumulative probability.

Figure 37.42 shows every 25th iteration and the last iteration for the Turnbull estimate of the CDF for the microprocessor data. The initial estimate assigns equal probabilities to each interval. You can specify different initial values with the PROBLIST= option. The algorithm converges in 130 iterations for this data. Convergence is determined if the change in the log-likelihood between two successive iterations less than Δ, where the default value is $\Delta = 10^{-8}$. You can specify a different value for delta with the TOLLIKE= option. This algorithm is an example of an expectation-maximization (EM) algorithm. EM algorithms are known to converge slowly, but the computations within each iteration for the Turnbull algorithm are moderate. Iterations will be terminated if the algorithm does not converge after a fixed number of iterations. The default maximum number of iterations is 1000. Some data may require more iterations for convergence. You can specify the maximum allowed number of iterations with the MAXITEM= option in the PROBPLOT, ANALYZE, or RPLOT statements.

The RELIABILITY Procedure

Iteration History for the Turnbull Estimate of the CDF

Iteration	Loglikelihood	(., 6)	(6, 12)	(24, 48)	(48, 168)
		(168, 500)	(500, 1000)	(1000, 2000)	(2000, .)
0	-1133.4051	0.125	0.125	0.125	0.125
		0.125	0.125	0.125	0.125
25	-104.16622	0.00421644	0.00140548	0.00140648	0.00173338
		0.00237846	0.00846094	0.04565407	0.93474475
50	-101.15151	0.00421644	0.00140548	0.00140648	0.00173293
		0.00234891	0.00727679	0.01174486	0.96986811
75	-101.06641	0.00421644	0.00140548	0.00140648	0.00173293
		0.00234891	0.00727127	0.00835638	0.9732621
100	-101.06534	0.00421644	0.00140548	0.00140648	0.00173293
		0.00234891	0.00727125	0.00801814	0.97360037
125	-101.06533	0.00421644	0.00140548	0.00140648	0.00173293
		0.00234891	0.00727125	0.00798438	0.97363413
130	-101.06533	0.00421644	0.00140548	0.00140648	0.00173293
		0.00234891	0.00727125	0.007983	0.97363551

Figure 37.42. Iteration History for Turnbull Estimate

If an interval probability is smaller than a tolerance (10^{-6} by default) after convergence, the probability is set to zero, the interval probabilities are renormalized so that they add to one, and iterations are restarted. Usually the algorithm converges in just a few more iterations. You can change the default value of the tolerance with the TOLPROB= option. You can specify the NOPOLISH option to avoid setting small probabilities to zero and restarting the algorithm.

If you specify the ITPRINTEM option, the table in Figure 37.43 summarizing the Turnbull estimate of the interval probabilities is printed. The columns labeled 'Reduced Gradient' and 'Lagrange Multiplier' are used in checking final convergence to the maximum likelihood estimate. The Lagrange multipliers must all be greater than or equal to zero, or the solution is not maximum likelihood. Refer to Gentleman and Geyer (1994) for more details of the convergence checking.

Lower Lifetime	Upper Lifetime	Probability	Reduced Gradient	Lagrange Multiplier
.	6	0.0042	0	0
6	12	0.0014	0	0
24	48	0.0014	0	0
48	168	0.0017	0	0
168	500	0.0023	0	0
500	1000	0.0073	-7.219342E-9	0
1000	2000	0.0080	-0.037063236	0
2000	.	0.9736	0.0003038877	0

Figure 37.43. Final Probability Estimates for Turnbull Algorithm

Figure 37.44 shows the final estimate of the CDF, along with standard errors and confidence limits. Figure 37.45 shows the CDF and simultaneous confidence limits plotted on a lognormal probability plot.

Cumulative Probability Estimates					
			Pointwise 95% Confidence Limits		
Lower Lifetime	Upper Lifetime	Cumulative Probability	Lower	Upper	Standard Error
6	6	0.0042	0.0019	0.0094	0.0017
12	24	0.0056	0.0028	0.0112	0.0020
48	48	0.0070	0.0038	0.0130	0.0022
168	168	0.0088	0.0047	0.0164	0.0028
500	500	0.0111	0.0058	0.0211	0.0037
1000	1000	0.0184	0.0094	0.0357	0.0063
2000	2000	0.0264	0.0124	0.0553	0.0101

Figure 37.44. Final CDF Estimates for Turnbull Algorithm

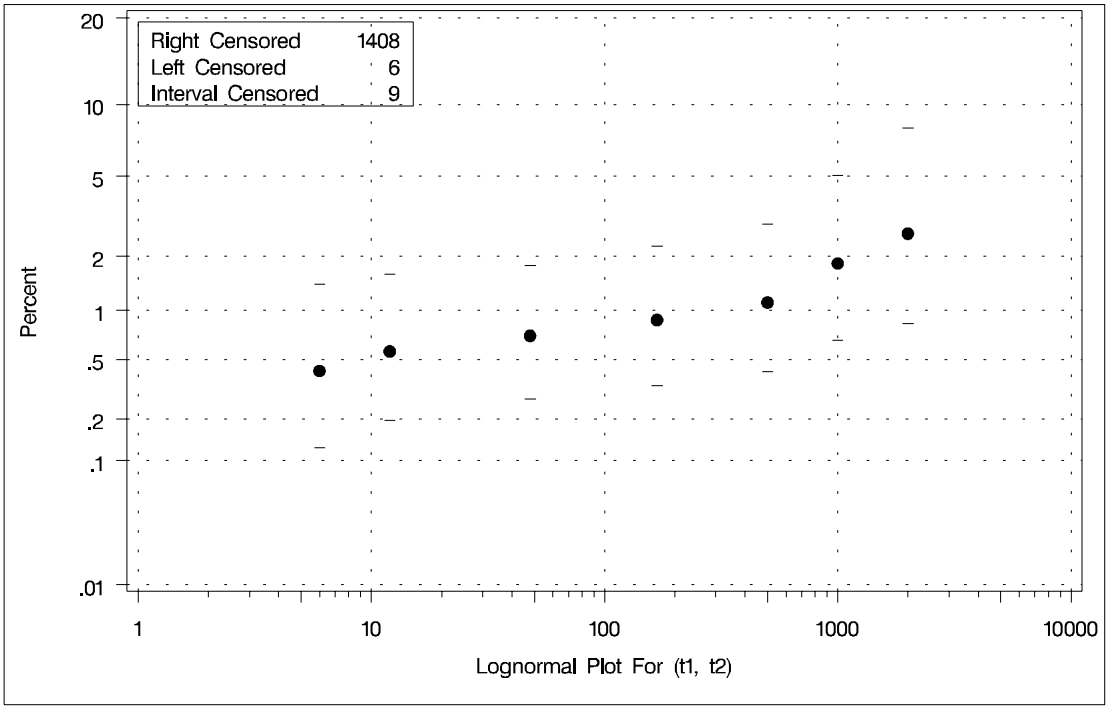


Figure 37.45. Lognormal Probability Plot for the Microprocessor Data

Nonparametric Confidence Intervals for Cumulative Failure Probabilities

The method used in the RELIABILITY procedure for computation of approximate pointwise and simultaneous confidence intervals for cumulative failure probabilities relies on the Kaplan-Meier estimator of the cumulative distribution function of failure time and approximate standard deviation of the Kaplan-Meier estimator. For the case of arbitrarily censored data, the Turnbull algorithm, discussed previously, provides an extension of the Kaplan-Meier estimator.

For multiply-censored data, the Kaplan-Meier estimator of the cumulative distribution function at failure time t_i is $\hat{F}(t_i) = 1 - \hat{S}(t_i)$, where

$$\hat{S}(t_i) = \prod_{j=1}^{i}(1 - \hat{p}_j),$$

$$\hat{p}_i = \frac{d_i}{n_i},$$

d_i is the number of failures in the interval (t_{i-1}, t_i) , and n_i is the number of unfailed units at the beginning of the interval. This definition of the Kaplan-Meier estimator is equivalent to the one previously given.

An estimator of the variance v_i of the Kaplan-Meier estimator $\hat{F}(t_i)$ is given by

$$\hat{v}_i = [\hat{S}(t_i)]^2 \sum_{j=1}^{i} \frac{\hat{p}_j}{n_j(1-\hat{p}_j)}$$

An estimator of the standard deviation of $\hat{F}(t_i)$ is $\text{se}_{\hat{F}} = \sqrt{\hat{v}_i}$.

For arbitrarily censored data, the Kaplan-Meier estimator is replaced by the nonparametric maximum likelihood estimator computed with the Turnbull algorithm, and the approximate variance of the estimator of $F(t_i)$ is computed from the inverse of the Fisher information matrix.

Pointwise Confidence Intervals

Approximate $(1-\alpha)100\%$ pointwise confidence intervals are computed as in Meeker and Escobar (1998, section 3.6) as

$$[F_L, F_U] = \left[\frac{\hat{F}}{\hat{F} + (1-\hat{F})w}, \quad \frac{\hat{F}}{\hat{F} + (1-\hat{F})/w} \right],$$

where

$$w = \exp\left[\frac{z_{1-\alpha/2}\text{se}_{\hat{F}}}{(\hat{F}(1-\hat{F}))} \right],$$

where z_p is the pth quantile of the standard normal distribution.

Simultaneous Confidence Intervals

Approximate $(1-\alpha)100\%$ simultaneous confidence bands valid over the lifetime interval (t_a, t_b) are computed as the "Equal Precision" case of Nair (1984) and Meeker and Escobar (1998, section 3.8) as

$$[F_L, F_U] = \left[\frac{\hat{F}}{\hat{F} + (1-\hat{F})w}, \quad \frac{\hat{F}}{\hat{F} + (1-\hat{F})/w} \right],$$

where

$$w = \exp\left[\frac{e_{a,b,1-\alpha/2}\text{se}_{\hat{F}}}{(\hat{F}(1-\hat{F}))} \right],$$

where the factor $x = e_{a,b,1-\alpha/2}$ is the solution of

$$x \exp(-x^2/2) \log\left[\frac{(1-a)b}{(1-b)a} \right] / \sqrt{8\pi} = \alpha/2$$

The time interval (t_a, t_b) over which the bands are valid depends in a complicated way on the constants a and b defined in Nair (1984), $0 < a < b < 1$. a and b are chosen by default, so that the confidence bands are valid between the lowest and highest times corresponding to failures in the case of multiply-censored data, or, to the lowest and highest intervals for which probabilities are computed for arbitrarily censored data. You can optionally specify a and b directly with the NPINTERVALS=SIMULTANEOUS(a,b) option in the PROBPLOT statement.

Parameter Estimation

Maximum Likelihood Estimation

Maximum likelihood estimation of the parameters of a statistical model involves maximizing the likelihood or, equivalently, the log likelihood with respect to the parameters. The parameter values at which the maximum occurs are the maximum likelihood estimates of the model parameters. The likelihood is a function of the parameters and of the data.

Let $x_1, x_2, \ldots, x_n$ be the observations in a random sample, including the failures and censoring times (if the data are censored). Let $f(\boldsymbol{\theta}; x)$ be the probability density of failure time, $S(\boldsymbol{\theta}; x) = Pr\{X \geq x\}$ be the reliability function, and $F(\boldsymbol{\theta}; x) = Pr\{X \leq x\}$ be the cumulative distribution function, where $\boldsymbol{\theta}$ is the vector of parameters to be estimated, $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_p)$. The probability density, reliability function, and CDF are determined by the specific distribution selected as a model for the data. The log likelihood is defined as

$$\begin{aligned} L(\boldsymbol{\theta}) &= \sum_i \log(f(\boldsymbol{\theta}; x_i)) + \sum_i{}' \log(S(\boldsymbol{\theta}; x_i)) + \\ &\quad \sum_i{}'' \log(F(\boldsymbol{\theta}; x_i)) + \sum_i{}''' [\log(F(\boldsymbol{\theta}; x_{ui}) - F(\boldsymbol{\theta}; x_{li}))] \end{aligned}$$

where

- $\sum$ is the sum over failed units
- $\sum'$ is the sum over right-censored units
- $\sum''$ is the sum over left-censored units
- $\sum'''$ is the sum over interval-censored units

and (x_{li}, x_{ui}) is the interval in which the ith unit is interval censored. Only the sums appropriate to the type of censoring in the data are included when the preceding equation is used.

The RELIABILITY procedure maximizes the log likelihood with respect to the parameters $\boldsymbol{\theta}$ using a Newton-Raphson algorithm. The Newton-Raphson algorithm is a recursive method for computing the maximum of a function. On the *r*th iteration, the algorithm updates the parameter vector $\boldsymbol{\theta}_r$ with

$$\boldsymbol{\theta}_{r+1} = \boldsymbol{\theta}_r - \mathbf{H}^{-1}\mathbf{g}$$

where $\mathbf{H}$ is the Hessian (second derivative) matrix, and $\mathbf{g}$ is the gradient (first derivative) vector of the log likelihood function, both evaluated at the current value of the parameter vector. That is,

$$\mathbf{g} = [g_j] = \left[\frac{\partial L}{\partial \theta_j}\right]\Bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}_r}$$

and

$$\mathbf{H} = [h_{ij}] = \left[\frac{\partial^2 L}{\partial \theta_i \partial \theta_j}\right]\Bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}_r}$$

Iteration continues until the parameter estimates converge. The convergence criterion is

$$|\theta_i^{r+1} - \theta_i^r| \leq c \quad \text{if} \quad |\theta_i^{r+1}| < .01$$

$$\left|\frac{\theta_i^{r+1} - \theta_i^r}{\theta_i^{r+1}}\right| \leq c \quad \text{if} \quad |\theta_i^{r+1}| \geq .01$$

for all $i = 1, 2, \ldots, p$ where c is the convergence criterion. The default value of c is 0.001, and it can be specified with the CONVERGE= option in the MODEL, PROBPLOT, RELATIONPLOT, and ANALYZE statements.

After convergence by the preceding criterion, the quantity

$$tc = \frac{\mathbf{g}\mathbf{H}^{-1}\mathbf{g}}{L}$$

is computed. If $tc > d$ then a warning is printed that the algorithm did not converge. tc is called the *relative Hessian* convergence criterion. The default value of d is .0001. You can specify other values for d with the CONVH= option. The relative Hessian criterion is useful in detecting the occasional case where no progress can be made in increasing the log-likelihood, yet the gradient **g** is not zero.

A location-scale model has a CDF of the form

$$F(x) = G\left(\frac{x - \mu}{\sigma}\right)$$

where μ is the location parameter, σ is the scale parameter, and G is a standardized form ($\mu = 0, \sigma = 1$) of the cumulative distribution function. The parameter vector is $\boldsymbol{\theta}$=(μ σ). It is more convenient computationally to maximize log likelihoods that arise from location-scale models. If you specify a distribution from Table 37.40 that is not a location-scale model, it is transformed to a location-scale model by taking the natural (base e) logarithm of the response. If you specify the lognormal base 10 distribution,

the logarithm (base 10) of the response is used. The Weibull, lognormal, and log-logistic distributions in Table 37.40 are not location-scale models. Table 37.41 shows the corresponding location-scale models that result from taking the logarithm of the response.

Maximum likelihood is the default method of estimating the location and scale parameters in the MODEL, PROBPLOT, RELATIONPLOT, and ANALYZE statements. If the Weibull distribution is specified, the logarithms of the responses are used to obtain maximum likelihood estimates ($\hat{\mu}$ $\hat{\sigma}$) of the location and scale parameters of the extreme value distribution. The maximum likelihood estimates ($\hat{\alpha}$, $\hat{\beta}$) of the Weibull scale and shape parameters are computed as $\hat{\alpha} = \exp(\hat{\mu})$ and $\hat{\beta} = 1/\hat{\sigma}$.

Regression Models

You can specify a regression model using the MODEL statement. For example, if you want to relate the lifetimes of electronic parts in a test to operating temperature using the Arrhenius relationship, then an appropriate model might be

$$\mu_i = \beta_0 + x_i\beta_1$$

where $x_i = 1000/(T_i + 273.15)$, and T_i is the centigrade temperature at which the ith unit is tested. Here, $\mathbf{x}_i' = [\ 1\ x_i\ ]$.

There are two types of explanatory variables: *continuous* variables and *class* (or *classification*) variables. Continuous variables represent physical quantities, such as temperature or voltage, and they must be numeric. Continuous explanatory variables are sometimes called *covariates*.

Class variables identify classification levels and are declared in the CLASS statement. These are also referred to as *categorical, dummy, qualitative, discrete,* or *nominal* variables. Class variables can be either character or numeric. The values of class variables are called *levels*. For example, the class variable BATCH could have levels 'batch1' and 'batch2' to identify items from two production batches. An indicator (0-1) variable is generated for each level of a class variable and is used as an explanatory variable. See Nelson (1990, p.277) for an example using an indicator variable in the analysis of accelerated life test data. In a model, an explanatory variable that is not declared in a CLASS statement is assumed to be continuous.

By default, all regression models automatically contain an intercept term; that is, the model is of the form

$$\mu_i = \beta_0 + \beta_1 x_{i1} + \ldots$$

where β_0 does not have an explanatory variable multiplier. The intercept term can be excluded from the model by specifying INTERCEPT$= 0$ as a MODEL statement option.

For numerical stability, continuous explanatory variables are centered and scaled internally to the procedure. This transforms the parameters $\boldsymbol{\beta}$ in the original model to a new set of parameters. The parameter estimates $\boldsymbol{\beta}$ and covariances are transformed back to the original scale before reporting, so that the parameters should be interpreted in terms of the originally specified model. Covariates that are indicator variables, that is, those specified in a CLASS statement, are not centered and scaled.

Initial values of the regression parameters used in the Newton-Raphson method are computed by ordinary least squares. The parameters $\boldsymbol{\beta}$ and the scale parameter σ are jointly estimated by maximum likelihood, taking a logarithmic transformation of the responses, if necessary, to get a location-scale model.

The generalized gamma distribution is fit using the log lifetimes. The regression parameters $\boldsymbol{\beta}$, the scale parameter σ, and the shape parameter λ are jointly estimated.

The Weibull distribution shape parameter estimate is computed as $\hat{\beta} = 1/\hat{\sigma}$, where σ is the scale parameter from the corresponding extreme value distribution. The Weibull scale parameter $\hat{\alpha}_i = \exp(\mathbf{x}_i{}'\hat{\boldsymbol{\beta}})$ is not computed by the procedure. Instead, the regression parameters $\boldsymbol{\beta}$ and the shape β are reported.

In a model with one to three continuous explanatory variables x, you can use the RELATION= option in the MODEL statement to specify a transformation that is applied to the variables before model fitting. Table 37.49 shows the available transformations.

Table 37.49. Variable Transformations

Relation	Transformed variable
ARRHENIUS (Nelson parameterization)	$1000/(x + 273.15)$
ARRHENIUS2 (activation energy parameterization)	$11605/(x + 273.15)$
POWER	$\log(x), \quad x > 0$
LINEAR	x
LOGISTIC	$\log\left(\frac{x}{1-x}\right), \quad 0 < x < 1$

Non-constant Scale Parameter

In some situations, it is desirable for the scale parameter to change with the values of explanatory variables. For example, Meeker and Escobar (1998, section 17.5) present an analysis of accelerated life test data where the spread of the data is greater at lower levels of the stress. You can use the LOGSCALE statement to specify the scale parameter as a function of explanatory variables. You must also have a MODEL statement to specify the location parameter. Explanatory variables can be continuous variables, indicator variables specified in the CLASS statement, or any interaction combination. The variables can be the same as specified in the MODEL statement, or they can be different variables. Any transformation specified with the RELATION= MODEL statement option will be applied to the same variable appearing in the LOGSCALE statement. See "Regression Model with Non-Constant Scale" on page 1109 for an example of fitting a model with non-constant scale parameter.

The form of the model for the scale parameter is

$$\log(\sigma_i) = \beta_0 + \beta_1 x_{i1} + \ldots$$

where β_0 is the intercept term. The intercept term can be excluded from the model by specifying INTERCEPT= 0 as a LOGSCALE statement option.

The parameters $\beta_0, \quad \beta_1, \quad , \ldots$ are estimated by maximum likelihood jointly with all the other parameters in the model.

Stable Parameters

The location and scale parameters (μ, σ) are estimated by maximizing the likelihood function by numerical methods, as described previously. An alternative parameterization that may have better numerical properties for heavy censoring is (η, σ), where $\eta = \mu + z_p\sigma$ and z_p is the pth quantile of the standardized distribution. See Meeker and Escobar (1998, p. 90) and Doganaksoy and Schmee (1993) for more details on alternate parameterizations.

By default, RELIABILITY estimates a value of z_p from the data that will improve the numerical properties of the estimation. You can also specify values of p from which the value of z_p will be computed with the PSTABLE= option in the ANALYZE, PROBPLOT, RELATIONPLOT, or MODEL statements. Note that a value of $p = 0.632$ for the Weibull and extreme value and $p = 0.5$ for all other distributions will give $z_p = 0$ and the parameterization will then be the usual location-scale parameterization.

All estimates and related statistics are reported in terms of the location and scale parameters (μ, σ). If you specify the ITPRINT option in the ANALYZE, PROBPLOT, or RELATIONPLOT statement, a table showing the values of p, ν, σ, and the last evaluation of the gradient and Hessian for these parameters is produced.

Covariance Matrix

An estimate of the covariance matrix of the maximum likelihood estimators (MLEs) of the parameters $\boldsymbol{\theta}$ is given by the inverse of the negative of the matrix of second derivatives of the log likelihood, evaluated at the final parameter estimates:

$$\boldsymbol{\Sigma} = [\sigma_{ij}] = -\mathbf{H}^{-1} = -\left[\frac{\partial^2 L}{\partial\theta_i \partial\theta_j}\right]^{-1}_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}$$

The negative of the matrix of second derivatives is called the Fisher information matrix. The diagonal term σ_{ii} is an estimate of the variance of $\hat{\theta}_i$. Estimates of standard errors of the MLEs are provided by

$$\mathrm{SE}_{\theta_i} = \sqrt{\sigma_{ii}}$$

An estimator of the correlation matrix is

$$\mathbf{R} = \left[\frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}}\right]$$

The covariance matrix for the Weibull distribution parameter estimators is computed by a first-order approximation from the covariance matrix of the estimators of the corresponding extreme value parameters (μ, σ) as

$$\begin{aligned}
\mathrm{Var}(\hat{\alpha}) &= [\exp(\hat{\mu})]^2 \mathrm{Var}(\hat{\mu}) \\
\mathrm{Var}(\hat{\beta}) &= \frac{\mathrm{Var}(\hat{\sigma})}{\hat{\sigma}^4} \\
\mathrm{Cov}(\hat{\alpha}, \hat{\beta}) &= -\frac{\exp(\hat{\mu})}{\hat{\sigma}^2}\mathrm{Cov}(\hat{\mu}, \hat{\sigma})
\end{aligned}$$

For the regression model, the variance of the Weibull shape parameter estimator $\hat{\beta}$ is computed from the variance of the estimator of the extreme value scale parameter σ as shown previously. The covariance of the regression parameter estimator $\hat{\beta}_i$ and the Weibull shape parameter estimator $\hat{\beta}$ is computed in terms of the covariance between $\hat{\beta}_i$ and $\hat{\sigma}$ as

$$\mathrm{Cov}(\hat{\beta}_i, \hat{\beta}) = -\frac{\mathrm{Cov}(\hat{\beta}_i, \hat{\sigma})}{\hat{\sigma}^2}$$

Confidence Intervals for Distribution Parameters

Table 37.50 shows the method of computation of approximate two-sided $\gamma \times 100\%$ confidence limits for distribution parameters. The default value of confidence is $\gamma = 0.95$. Other values of confidence are specified using the CONFIDENCE= option. In Table 37.50, K_γ represents the $(1+\gamma)/2 \times 100\%$ percentile of the standard normal distribution, and $\hat{\mu}$ and $\hat{\sigma}$ are the MLEs of the location and scale parameters for the normal, extreme value, and logistic distributions. For the lognormal, Weibull, and log-logistic distributions, $\hat{\mu}$ and $\hat{\sigma}$ represent the MLEs of the corresponding location and scale parameters of the location-scale distribution that results when the logarithm of the lifetime is used as the response. For the Weibull distribution, μ and σ are the location and scale parameters of the extreme value distribution for the logarithm of the lifetime. $\mathrm{SE}_{\hat{\theta}}$ denotes the standard error of the MLE of θ, computed as the square root of the appropriate diagonal element of the inverse of the Fisher information matrix.

Regression Parameters

Approximate $\gamma \times 100\%$ confidence limits for the regression parameter β_i are given by

$$\beta_{iL} = \hat{\beta}_i - K_\gamma(\mathrm{SE}_{\hat{\beta}_i})$$

$$\beta_{iU} = \hat{\beta}_i + K_\gamma(\mathrm{SE}_{\hat{\beta}_i})$$

Table 37.50. Confidence Limit Computation

Distribution	Parameters		
	Location	Scale	Shape
Normal	$\mu_L = \hat{\mu} - K_\gamma(\mathrm{SE}_{\hat{\mu}})$ $\mu_U = \hat{\mu} + K_\gamma(\mathrm{SE}_{\hat{\mu}})$	$\sigma_L = \hat{\sigma}/\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$ $\sigma_U = \hat{\sigma}\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$	
Lognormal	$\mu_L = \hat{\mu} - K_\gamma(\mathrm{SE}_{\hat{\mu}})$ $\mu_U = \hat{\mu} + K_\gamma(\mathrm{SE}_{\hat{\mu}})$	$\sigma_L = \hat{\sigma}/\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$ $\sigma_U = \hat{\sigma}\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$	
Lognormal (base 10)	$\mu_L = \hat{\mu} - K_\gamma(\mathrm{SE}_{\hat{\mu}})$ $\mu_U = \hat{\mu} + K_\gamma(\mathrm{SE}_{\hat{\mu}})$	$\sigma_L = \hat{\sigma}/\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$ $\sigma_U = \hat{\sigma}\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$	
Extreme Value	$\mu_L = \hat{\mu} - K_\gamma(\mathrm{SE}_{\hat{\mu}})$ $\mu_U = \hat{\mu} + K_\gamma(\mathrm{SE}_{\hat{\mu}})$	$\sigma_L = \hat{\sigma}/\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$ $\sigma_U = \hat{\sigma}\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$	
Weibull		$\alpha_L = \exp[\hat{\mu} - K_\gamma(\mathrm{SE}_{\hat{\mu}})]$ $\alpha_U = \exp[\hat{\mu} + K_\gamma(\mathrm{SE}_{\hat{\mu}})]$	$\beta_L = \exp[-K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]/\hat{\sigma}$ $\beta_U = \exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]/\hat{\sigma}$
Exponential		$\alpha_L = \exp[\hat{\mu} - K_\gamma(\mathrm{SE}_{\hat{\mu}})]$ $\alpha_U = \exp[\hat{\mu} + K_\gamma(\mathrm{SE}_{\hat{\mu}})]$	
Logistic	$\mu_L = \hat{\mu} - K_\gamma(\mathrm{SE}_{\hat{\mu}})$ $\mu_U = \hat{\mu} + K_\gamma(\mathrm{SE}_{\hat{\mu}})$	$\sigma_L = \hat{\sigma}/\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$ $\sigma_U = \hat{\sigma}\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$	
Log-logistic	$\mu_L = \hat{\mu} - K_\gamma(\mathrm{SE}_{\hat{\mu}})$ $\mu_U = \hat{\mu} + K_\gamma(\mathrm{SE}_{\hat{\mu}})$	$\sigma_L = \hat{\sigma}/\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$ $\sigma_U = \hat{\sigma}\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$	
Generalized gamma		$\sigma_L = \hat{\sigma}/\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$ $\sigma_U = \hat{\sigma}\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$	$\mu_L = \hat{\lambda} - K_\gamma(\mathrm{SE}_{\hat{\lambda}})$ $\mu_U = \hat{\lambda} + K_\gamma(\mathrm{SE}_{\hat{\lambda}})$

Percentiles

The maximum likelihood estimate of the $p \times 100\%$ percentile x_p for the extreme value, normal, and logistic distributions is given by

$$\hat{x}_p = \hat{\mu} + z_p\hat{\sigma}$$

where $z_p = G^{-1}(p)$, G is the standardized CDF shown in Table 37.51, and $(\hat{\mu}, \hat{\sigma})$ are the maximum likelihood estimates of the location and scale parameters of the distribution. The maximum likelihood estimate of the percentile t_p for the Weibull, lognormal, and log-logistic distributions is given by

$$\hat{t}_p = \exp[\hat{\mu} + z_p\hat{\sigma}]$$

where $z_p = G^{-1}(p)$, and G is the standardized CDF of the location-scale model corresponding to the logarithm of the response. For the lognormal (base 10) distribution,

$$\hat{t}_p = 10^{[\hat{\mu}+z_p\hat{\sigma}]}$$

Table 37.51. Standardized Cumulative Distribution Functions

Distribution	Location-Scale Distribution	Location-Scale CDF
Weibull	Extreme Value	$1 - \exp[-\exp(z)]$
Lognormal	Normal	$\int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right) du$
Log-logistic	Logistic	$\frac{\exp(z)}{1+\exp(z)}$

Confidence Intervals

The variance of the MLE of the $p \times 100\%$ percentile for the normal, extreme value, or logistic distribution is

$$Var(\hat{x}_p) = Var(\hat{\mu}) + z_p^2 Var(\hat{\sigma}) + 2Cov(\hat{\mu}, \hat{\sigma})$$

Two-sided approximate $100\gamma\%$ confidence limits for x_p are

$$\begin{aligned} x_{pL} &= \hat{x}_p - K_\gamma\sqrt{Var(\hat{x}_p)} \\ x_{pU} &= \hat{x}_p + K_\gamma\sqrt{Var(\hat{x}_p)} \end{aligned}$$

where K_γ represents the $100(1+\gamma)/2 \times 100\%$ percentile of the standard normal distribution.

The limits for the lognormal, Weibull, or log-logistic distributions are

$$\begin{aligned} t_{pL} &= \exp\left(\hat{x}_p - K_\gamma\sqrt{Var(\hat{x}_p)}\right) \\ t_{pU} &= \exp\left(\hat{x}_p + K_\gamma\sqrt{Var(\hat{x}_p)}\right) \end{aligned}$$

where x_p refers to the percentile of the corresponding location-scale distribution (normal, extreme value, or logistic) for the logarithm of the lifetime. For the lognormal (base 10) distribution,

$$\begin{aligned} t_{pL} &= 10^{\left(\hat{x}_p - K_\gamma\sqrt{Var(\hat{x}_p)}\right)} \\ t_{pU} &= 10^{\left(\hat{x}_p + K_\gamma\sqrt{Var(\hat{x}_p)}\right)} \end{aligned}$$

Reliability Function

For the extreme value, normal, and logistic distributions shown in Table 37.51, the maximum likelihood estimate of the reliability function $R(x) = Pr\{X > x\}$ is given by

$$\hat{R}(x) = 1 - F\left(\frac{x-\hat{\mu}}{\hat{\sigma}}\right)$$

The MLE of the CDF is $\hat{F}(x) = 1 - \hat{R}(x)$.

Confidence Intervals

Let $\hat{u} = \frac{x-\hat{\mu}}{\hat{\sigma}}$. The variance of u is

$$Var(\hat{u}) \approx \frac{Var(\hat{\mu}) + \hat{u}^2 Var(\hat{\sigma}) + 2\hat{u}Cov(\hat{\mu},\hat{\sigma})}{\hat{\sigma}^2}$$

Two-sided approximate $\gamma \times 100\%$ confidence intervals for $R(x)$ are computed as

$$R_L(x) = \hat{R}(u_2)$$

$$R_U(x) = \hat{R}(u_1)$$

where

$$u_1 = \hat{u} - K_\gamma\sqrt{Var(\hat{u})}$$

$$u_2 = \hat{u} + K_\gamma\sqrt{Var(\hat{u})}$$

and K_γ represents the $(1+\gamma)/2\times 100\%$ percentile of the standard normal distribution.

The corresponding limits for the CDF are

$$F_L(x) = 1 - R_U(x)$$

$$F_U(x) = 1 - R_L(x)$$

Limits for the Weibull, lognormal, and log-logistic reliability function $R(t)$ are the same as those for the corresponding extreme value, normal, or logistic reliability $R(y)$, where $y = \log(t)$.

Estimation with the Binomial and Poisson Distributions

In addition to estimating the parameters of the distributions in Table 37.40, you can estimate parameters, compute confidence limits, compute predicted values and prediction limits, and compute chi-squared tests for differences in groups for the binomial and Poisson distributions using the ANALYZE statement. Specify either BINOMIAL or POISSON in the DISTRIBUTION statement to use one of these distributions. The ANALYZE statement options available for the binomial and Poisson distributions are given in Table 37.6. See "Analysis of Binomial Data" on page 1129 for an example of an analysis of binomial data.

Binomial Distribution

If r is the number of successes and n is the number of trials in a binomial experiment, then the maximum likelihood estimator of the probability p in the binomial distribution in Table 37.42 is computed as

$$\hat{p} = r/n$$

Two-sided $\gamma \times 100\%$ confidence limits for p are computed as in Johnson, Kotz, and Kemp (1992, p.130):

$$p_L = \frac{\nu_1 F[(1-\gamma)/2; \nu_1, \nu_2]}{\nu_2 + \nu_1 F[(1-\gamma)/2; \nu_1, \nu_2]}$$

with $\nu_1 = 2r$ and $\nu_2 = 2(n - r + 1)$ and

$$p_U = \frac{\nu_1 F[(1+\gamma)/2; \nu_1, \nu_2]}{\nu_2 + \nu_1 F[(1+\gamma)/2; \nu_1, \nu_2]}$$

with $\nu_1 = 2(r+1)$ and $\nu_2 = 2(n-r)$, where $F[\gamma; \nu_1, \nu_2]$ is the $\gamma \times 100\%$ percentile of the F distribution with ν_1 degrees of freedom in the numerator and ν_2 degrees of freedom in the denominator.

You can compute a sample size required to estimate p within a specified tolerance w with probability γ. Nelson (1982, p. 206) gives the following formula for the approximate sample size:

$$n \approx \hat{p}(1-\hat{p})\left(\frac{K_\gamma}{w}\right)^2$$

where K_γ is the $(1+\gamma)/2 \times 100\%$ percentile of the standard normal distribution. The formula is based on the normal approximation for the distribution of $\hat{p}$. Nelson recommends using this formula if $np > 10$ and $np(1-p) > 10$. The value of γ used for computing confidence limits is used in the sample size computation. The default value of confidence is $\gamma = 0.95$. Other values of confidence are specified using the CONFIDENCE= option. You specify a tolerance of *number* with the TOLERANCE(*number*) option.

The predicted number of successes X in a future sample of size m, based on the previous estimate of p, is computed as

$$\hat{X} = m(r/n) = m\hat{p}$$

Two-sided approximate $\gamma \times 100)\%$ prediction limits are computed as in Nelson (1982, p. 208). The prediction limits are the solutions X_L and X_U of

$$X_U/m = [(r+1)/n]F[(1+\gamma)/2; 2(r+1), 2X_U]$$

$$m/(X_L+1) = (n/r)F[(1+\gamma)/2; 2(X_L+1), 2r]$$

where $F[\gamma; \nu_1, \nu_2]$ is the $\gamma \times 100\%$ percentile of the F distribution with ν_1 degrees of freedom in the numerator and ν_2 degrees of freedom in the denominator. You request predicted values and prediction limits for a future sample of size *number* with the PREDICT(*number*) option.

You can test groups of binomial data for equality of their binomial probability using the ANALYZE statement. You specify the K groups to be compared with a group variable having K levels.

Nelson (1982, p.450) discusses a chi-squared test statistic for comparing K binomial proportions for equality. Suppose there are r_i successes in n_i trials for $i = 1, 2, \ldots, K$. The grouped estimate of the binomial probability is

$$\hat{p} = \frac{r_1 + r_2 + \cdots + r_K}{n_1 + n_2 + \cdots + n_K}$$

The chi-squared test statistic for testing the hypothesis $p_1 = p_2 = \ldots = p_K$ against $p_i \neq p_j$ for some i and j is

$$Q = \sum_{i=1}^{K} \frac{(r_i - n_i\hat{p})^2}{n_i\hat{p}(1-\hat{p})}$$

The statistic Q has an asymptotic chi-squared distribution with $K-1$ degrees of freedom. The RELIABILITY procedure computes the contribution of each group to Q, the value of Q, and the *p*-value for Q based on the limiting chi-squared distribution with $K-1$ degrees of freedom. If you specify the PREDICT option, predicted values and prediction limits are computed for each group, as well as for the pooled group. The *p*-value is defined as $p_0 = 1 - \chi^2_{K-1}[Q]$, where $\chi^2_{K-1}[x]$ is the chi-squared CDF with $K-1$ degrees of freedom, and Q is the observed value. A test of the hypothesis of equal binomial probabilities among the groups with significance level α is

- $p_0 > \alpha$: do not reject the equality hypothesis
- $p_0 \leq \alpha$: reject the equality hypothesis

Poisson Distribution

You can use the ANALYZE statement to model data using the Poisson distribution. The data consists of a count Y of occurrences in a "length" of observation T. Observation T is typically an *exposure time*, but it can have other units, such as distance. The ANALYZE statement enables you to compute the rate of occurrences, confidence limits, and prediction limits.

An estimate of the rate λ is computed as

$$\hat{\lambda} = Y/T$$

Two-sided $\gamma \times 100\%$ confidence limits for λ are computed as in Nelson (1982, p. 201):

$$\lambda_L = .5\chi^2[(1-\gamma)/2; 2Y]/T$$

$$\lambda_U = .5\chi^2[(1+\gamma)/2; 2(Y+1)]/T$$

where $\chi^2[\delta; \nu]$ is the $\delta \times 100\%$ percentile of the chi-squared distribution with ν degrees of freedom.

You can compute a length T required to estimate λ within a specified tolerance w with probability γ. Nelson (1982, p. 202) provides the following approximate formula:

$$\hat{T} \approx \hat{\lambda} \left(\frac{K_\gamma}{w} \right)^2$$

where K_γ is the $(1+\gamma)/2 \times 100\%$ percentile of the standard normal distribution. The formula is based on the normal approximation for $\hat{\lambda}$ and is more accurate for larger values of λT. Nelson recommends using the formula when $\lambda T > 10$. The value of γ used for computing confidence limits is also used in the length computation. The default value of confidence is $\gamma = 0.95$. Other values of confidence are specified using the CONFIDENCE= option. You specify a tolerance of *number* with the TOLERANCE(*number*) option.

The predicted future number of occurrences in a length S is

$$\hat{X} = (Y/T)S = \hat{\lambda}S$$

Two-sided approximate $\gamma \times 100\%$ prediction limits are computed as in Nelson (1982, p. 203). The prediction limits are the solutions X_L and X_U of

$$X_U/S = [(Y+1)/T]F[(1+\gamma)/2; 2(Y+1), 2X_U]$$

$$S/(X_L+1) = (T/Y)F[(1+\gamma)/2; 2(X_L+1), 2Y]$$

where $F[\gamma; \nu_1, \nu_2]$ is the $\gamma \times 100\%$ percentile of the F distribution with ν_1 degrees of freedom in the numerator and ν_2 degrees of freedom in the denominator. You request predicted values and prediction limits for a future exposure *number* with the PREDICT(*number*) option.

You can compute a chi-squared test statistic for comparing K Poisson rates for equality. You specify the K groups to be compared with a group variable having K levels.

Refer to Nelson (1982, p.444) for more information on this test. Suppose that there are Y_i Poisson counts in lengths T_i for $i = 1, 2, \ldots, K$ and that the Y_i are independent. The grouped estimate of the Poisson rate is

$$\hat{\lambda} = \frac{Y_1 + Y_2 + \cdots + Y_K}{T_1 + T_2 + \cdots + T_K}$$

The chi-squared test statistic for testing the hypothesis $\lambda_1 = \lambda_2 = \ldots = \lambda_K$ against $\lambda_i \neq \lambda_j$ for some i and j is

$$Q = \sum_{i=1}^{K} \frac{(Y_i - \hat{\lambda}T_i)^2}{\hat{\lambda}T_i}$$

The statistic Q has an asymptotic chi-squared distribution with $K-1$ degrees of freedom. The RELIABILITY procedure computes the contribution of each group to Q, the value of Q, and the *p*-value for Q based on the limiting chi-squared distribution with $K-1$ degrees of freedom. If you specify the PREDICT option, predicted values and prediction limits are computed for each group, as well as for the pooled group. The *p*-value is defined as $p_0 = 1 - \chi^2_{K-1}[Q]$, where $\chi^2_{K-1}[x]$ is the chi-squared CDF with $K-1$ degrees of freedom and Q is the observed value. A test of the hypothesis of equal Poisson rates among the groups with significance level α is

- $p_0 > \alpha$: accept the equality hypothesis
- $p_0 \leq \alpha$: reject the equality hypothesis

Least Squares Fit to the Probability Plot

Fitting to the probability plot by least squares is an alternative to maximum likelihood estimation of the parameters of a life distribution. Only the failure times are used. A least squares fit is computed using points $(x_{(i)}, m_i)$, where $m_i = F^{-1}(a_i)$ and a_i are the plotting positions as defined in "Probability Plotting" on page 1177. The x_i are either the lifetimes for the normal, extreme value, or logistic distributions or the log lifetimes for the lognormal, Weibull, or log-logistic distributions. The ANALYZE, PROBPLOT, or RELATIONPLOT statement option FITTYPE=LSXY specifies the $x_{(i)}$ as the dependent variable ('y-coordinate') and the m_i as the independent variable ('x-coordinate'). You can optionally reverse the quantities used as dependent and independent variables by specifying the FITTYPE=LSYX option.

Weibayes Estimation

Weibayes estimation is a method of performing a Weibull analysis when there are few or no failures. The FITTYPE=WEIBAYES option requests this method. The method of Nelson (1985) is used to compute a one-sided confidence interval for the Weibull scale parameter when the Weibull shape parameter is specified. Also refer to Abernethy (1996) for more discussion and examples. The Weibull shape parameter β is assumed to be known and is specified to the procedure with the SHAPE=*number* option. Let $T_1, T_2, \ldots, T_n$ be the failure and censoring times, and let $r \geq 0$ be the number of failures in the data. If there are no failures $(r = 0)$, a lower $\gamma \times 100\%$ confidence limit for the Weibull scale parameter α is computed as

$$\alpha_L = \{\sum_{i=1}^{n} T_i^{\beta} / [-\log(1-\gamma)]\}^{1/\beta}$$

The default value of confidence is $\gamma = 0.95$. Other values of confidence are specified using the CONFIDENCE= option.

If $r \geq 1$, the MLE of α is given by

$$\hat{\alpha} = [\sum_{i=1}^{n} T_i^{\beta} / r]^{1/\beta}$$

and a lower $\gamma \times 100\%$ confidence limit for the Weibull scale parameter α is computed as

$$\alpha_L = \hat{\alpha}[2r / \chi^2(\gamma, 2r+2)]^{1/\beta}$$

where $\chi^2(\gamma, 2r+2)$ is the γ percentile of a chi-square distribution with $2r+2$ degrees of freedom. The procedure uses the specified value of β and the computed value of α_L to compute distribution percentiles and the reliability function.

Estimation With Multiple Failure Modes

In many applications, units can experience multiple causes of failure, or *failure modes*. For example, in "Weibull Probability Plot for Two Combined Failure Modes" on page 1115, insulation specimens can either experience early failures due to manufacturing defects, or degradation failures due to aging. The FMODE statement is used to analyze this type of data. See "FMODE Statement" on page 1139 for the syntax of the FMODE statement. This section describes the analysis of data when units experience multiple failure modes. The assumptions used in the analysis are:

- a cause, or mode, can be identified for each failure
- failure modes follow a series-system model, i.e., a unit fails when a failure due to one of the modes occurs
- each failure mode has the specified lifetime distribution with different parameters
- failure modes act statistically independently

Suppose there are m failure modes, with lifetime distribution functions $F_1(t), F_2(t), \ldots, F_m(t)$.

If you wish to estimate the lifetime distribution of a failure mode, say mode i, acting alone, specify the KEEP keyword in the FMODE statement. The failures from all other modes are treated as right-censored observations, and the lifetime distribution is estimated by one of the methods described in other sections, for example, maximum likelihood. This lifetime distribution is interpreted as the distribution if the specified failure mode is acting alone, with all other modes eliminated. You can also specify more than one mode to KEEP, but the assumption is that all the specified modes have the same distribution.

If you specify the ELIMINATE keyword, failures due to the specified modes are treated as right-censored. The resulting distribution estimate is the failure distribution if the specified modes are eliminated.

If you specify the COMBINE keyword, the failure distribution when all the modes specified in the FMODE statement modes act is estimated. The failure distribution $F_i(t), i = 1, 2, \ldots, m$ from each individual mode is first estimated by treating all failures from other modes as right-censored. The estimated failure distributions are then combined to get an estimate of the lifetime distribution when all modes act

$$\hat{F}(t) = 1 - \prod_{i=1}^{m} [1 - \hat{F}_i(t)]$$

Pointwise approximate asymptotic normal confidence limits for $F(t)$ can be obtained by the delta method. See Meeker and Escobar (1998, Appendix B.2). The delta method variance of $\hat{F}(t)$ is, using the independence of failure modes,

$$Var(\hat{F}(t)) = \sum_{i=1}^{m} [S_0(u_1) S_0(u_2) \ldots f_0(u_i) \ldots S_0(u_m)]^2 Var(u_i)$$

where $u_i = \frac{y-\hat{\mu}_i}{\hat{\sigma}_i}$, y is t for the extreme value, normal , and logistic distributions or $\log(t)$ for the Weibull, lognormal or loglogistic distributions, $\hat{\mu}_i$ and $\hat{\sigma}_i$ are location and scale parameter estimates for mode i, and S_0 and f_0 are the standard ($\mu = 0, \sigma = 1$) survival function and density function for the specified distribution.

Two-sided approximate $(1-\alpha)100\%$ pointwise confidence intervals are computed as in Meeker and Escobar (1998, section 3.6) as

$$[F_L, F_U] = \left[\frac{\hat{F}}{\hat{F} + (1-\hat{F})w}, \quad \frac{\hat{F}}{\hat{F} + (1-\hat{F})/w} \right],$$

where

$$w = \exp\left[\frac{z_{1-\alpha/2}\text{se}_{\hat{F}}}{(\hat{F}(1-\hat{F}))} \right],$$

where $\text{se}_{\hat{F}} = \sqrt{Var(\hat{F}(t))}$ and z_p is the pth quantile of the standard normal distribution.

Regression Model Observation-Wise Statistics

For regression models that are fit using the MODEL statement, you can specify a variety of statistics to be computed for each observation in the input data set. This section describes the method of computation for each statistic. See Table 37.24 and Table 37.25 for the syntax for requesting these statistics.

Predicted Values

The linear predictor is

$$\hat{\mu}_i = \mathbf{x}_i'\boldsymbol{\beta}$$

where $\mathbf{x}_i$ is the vector of explanatory variables for the ith observation.

Percentiles

An estimator of the $p \times 100\%$ percentile x_p for the ith observation for the extreme value, normal, and logistic distributions is

$$\hat{x}_{i,p} = \mathbf{x}_i'\hat{\boldsymbol{\beta}} + z_p\hat{\sigma}$$

where $z_p = G^{-1}(p)$, G is the standardized CDF, and σ is the distribution scale parameter.

An estimator of the $p \times 100\%$ percentile t_p for the ith observation for the Weibull, lognormal, and log-logistic distributions is

$$\hat{t}_{i,p} = \exp[\mathbf{x}_i'\hat{\boldsymbol{\beta}} + z_p\hat{\sigma}]$$

where G is the standardized CDF of the extreme value, normal, or logistic distribution that corresponds to the logarithm of the lifetime, and σ is the distribution scale parameter.

The percentile of the lognormal (base 10) distribution is

$$\hat{t}_{i,p} = 10^{[\mathbf{x}_i{}'\hat{\boldsymbol{\beta}}+z_p\hat{\sigma}]}$$

where G is the CDF of the standard normal distribution.

An estimator of the $p \times 100\%$ percentile t_p for the ith observation for the generalized gamma distribution is

$$\hat{t}_{i,p} = \exp[\mathbf{x}_i{}'\hat{\boldsymbol{\beta}} + w_{\lambda,p}\hat{\sigma}]$$

where

$$w_{\lambda,p} = \frac{1}{\lambda}\log\left(\frac{\lambda^2}{2}\chi^2_{(2/\lambda^2),p}\right)$$

and $\chi^2_{k,p}$ is the $p \times 100\%$ percentile of the chi-squared distribution with k degrees of freedom.

Standard Errors of Percentile Estimator

For the extreme value, normal, and logistic distributions, the standard error of the estimator of the $p \times 100\%$ percentile is computed as

$$\sigma_{i,p} = \sqrt{\mathbf{z}'\boldsymbol{\Sigma}\mathbf{z}}$$

where

$$\mathbf{z} = \begin{bmatrix} \mathbf{x}_i \\ z_p \end{bmatrix}$$

and $\boldsymbol{\Sigma}$ is the covariance matrix of $(\hat{\boldsymbol{\beta}}, \hat{\sigma})$.

For the Weibull, lognormal, and log-logistic distributions, the standard error is computed as

$$\sigma_{i,p} = \exp(x_{i,p})\sqrt{\mathbf{z}'\boldsymbol{\Sigma}\mathbf{z}}$$

where $x_{i,p}$ is the percentile computed from the extreme value, normal, or logistic distribution that corresponds to the logarithm of the lifetime. The standard error for the lognormal (base 10) distribution is computed as

$$\sigma_{i,p} = 10^{x_{i,p}}\sqrt{\mathbf{z}'\boldsymbol{\Sigma}\mathbf{z}}$$

The standard error for the generalized gamma distribution percentile is computed as

$$\sigma_{i,p} = \exp[\mathbf{x}_i{}'\hat{\boldsymbol{\beta}} + w_{\lambda,p}\hat{\sigma}]\sqrt{\mathbf{z}'\boldsymbol{\Sigma}\mathbf{z}}$$

where

$$\mathbf{z} = \begin{bmatrix} \mathbf{x}_i \\ w_{\lambda,p} \\ \hat{\sigma}\frac{\partial w_{\lambda,p}}{\partial \lambda} \end{bmatrix}$$

$\boldsymbol{\Sigma}$ is the covariance matrix of $(\hat{\boldsymbol{\beta}}, \hat{\sigma}, \hat{\lambda})$, $\boldsymbol{\beta}$ is the vector of regression parameters, σ is the scale parameter, and λ is the shape parameter.

Confidence Limits for Percentiles

Two-sided approximate $100\gamma\%$ confidence limits for $x_{i,p}$ for the extreme value, normal, and logistic distributions are computed as

$$\begin{aligned} x_L &= \hat{x}_{i,p} - K_\gamma\sigma_{i,p} \\ x_U &= \hat{x}_{i,p} + K_\gamma\sigma_{i,p} \end{aligned}$$

where K_γ represents the $100(1+\gamma)/2 \times 100\%$ percentile of the standard normal distribution.

Limits for the Weibull, lognormal, and log-logistic percentiles are computed as

$$\begin{aligned} t_L &= \exp(x_L) \\ t_U &= \exp(x_U) \end{aligned}$$

where x_L and x_U are computed from the corresponding distributions for the logarithms of the lifetimes. For the lognormal (base 10) distribution,

$$\begin{aligned} t_L &= 10^{x_L} \\ t_U &= 10^{x_U} \end{aligned}$$

Limits for the generalized gamma distribution percentiles are computed as

$$t_L = \exp\left[\mathbf{x}_i{}'\hat{\boldsymbol{\beta}} + w_{\lambda,p}\hat{\sigma} - K_\gamma\sqrt{\mathbf{z}'\boldsymbol{\Sigma}\mathbf{z}}\right]$$
$$t_U = \exp\left[\mathbf{x}_i{}'\hat{\boldsymbol{\beta}} + w_{\lambda,p}\hat{\sigma} + K_\gamma\sqrt{\mathbf{z}'\boldsymbol{\Sigma}\mathbf{z}}\right]$$

Reliability Function

For the extreme value, normal, and logistic distributions, an estimate of the reliability function evaluated at the response y_i is computed as

$$R(y_i) = 1 - G\left(\frac{y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}}{\hat{\sigma}}\right)$$

where $G(x)$ is the standardized CDF of the distribution from Table 37.51.

Estimates of the reliability function evaluated at the response t_i for the Weibull, lognormal, log-logistic, and generalized gamma distributions are computed as

$$R(t_i) = 1 - G\left(\frac{\log(t_i) - \mathbf{x}_i'\hat{\boldsymbol{\beta}}}{\hat{\sigma}}\right)$$

where $G(x)$ is the standardized CDF of the corresponding extreme value, normal, logistic, or generalized log-gamma distributions.

Residuals

The RELIABILITY procedure computes several different kinds of residuals. In the following equations, y_i represents the ith response value if the extreme value, normal, or logistic distributions are specified. If t_i is the ith response and if the Weibull, lognormal, log-logistic, or generalized gamma distributions are specified, then y_i represents the logarithm of the response $y_i = \log(t_i)$. If the lognormal (base 10) distribution is specified, then $y_i = \log_{10}(t_i)$.

Raw Residuals

The raw residual is computed as

$$r_{Ri} = y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}$$

Standardized Residuals

The standardized residual is computed as

$$r_{Si} = \frac{y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}}{\hat{\sigma}}$$

Adjusted Residuals

If an observation is right censored, then the standardized residual for that observation is also right censored. Adjusted residuals adjust censored standardized residuals upward by adding a percentile of the residual lifetime distribution, given that the standardized residual exceeds the censoring value. The default percentile is the median (50th percentile), but you can, optionally, specify a $\gamma \times 100\%$ percentile using the

RESIDALPHA=γ option in MODEL statement. The $\gamma \times 100$ percentile residual life is computed as in Joe and Proschan (1984). The adjusted residual is computed as

$$r_{Ai} = \begin{cases} G^{-1}[1-(1-\gamma)S(u_i)] & \text{for right-censored observations} \\ u_i & \text{for uncensored observations} \end{cases}$$

where G is the standard CDF,

$$S(u) = 1 - G(u)$$

is the reliability function, and

$$u_i = \frac{y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}}{\hat{\sigma}}$$

If the generalized gamma distribution is specified, the standardized CDF and reliability functions include the estimated shape parameter $\hat{\lambda}$.

Modified Cox-Snell Residuals

Let

$$\delta_i = \begin{cases} 1 & \text{for uncensored observations} \\ 0 & \text{for right-censored observations} \end{cases}$$

The Cox-Snell residual is defined as

$$r_{Ci} = -\log(R(y_i))$$

where

$$R(y) = 1 - G\left(\frac{y - \mathbf{x}_i'\hat{\boldsymbol{\beta}}}{\hat{\sigma}}\right)$$

is the reliability function. The modified Cox-Snell residual is computed as in Collett (1994, p.152):

$$r_{Ci}^{'} = r_{Ci} + (1-\delta_i)\alpha$$

where α is an adjustment factor. If the fitted model is correct, the Cox-Snell residual has approximately a standard exponential distribution for uncensored observations. If an observation is censored, the residual evaluated at the censoring time is not as large as the residual evaluated at the (unknown) failure time. The adjustment factor α adjusts the censored residuals upward to account for the censoring. The default is $\alpha = 0.693$, the median of the standard exponential distribution. You can, optionally, specify any adjustment factor by using the MODEL statement option RESIDADJ=α. Another commonly used value is the mean of the standard exponential distribution, $\alpha = 1$.

Deviance Residuals

Deviance residuals are a zero-mean, symmetrized version of modified Cox-Snell residuals. Deviance residuals are computed as in Collett (1994, p.153):

$$r_{Di} = \text{sgn}(\delta_i - r_{Ci})\{-2[\delta_i - r_{Ci} + \delta_i \log(r_{Ci})]\}^{1/2}$$

where

$$\text{sgn}(u) = \begin{cases} -1 & \text{if } u < 0 \\ 1 & \text{if } u \geq 0 \end{cases}$$

Recurrence Data from Repairable Systems

When a repairable system fails, it is repaired and placed back in service. As a repairable system ages, it accumulates a history of repairs and costs of repairs. The mean cumulative function (MCF) $M(t)$ is defined as the population mean of the cumulative number (or cost) of repairs up until time t. You can use the RELIABILITY procedure to compute and plot nonparametric estimates and plots of the MCF for the number of repairs or the cost of repairs. The Nelson (1995) confidence limits for the MCF are also computed and plotted. You can compute and plot estimates of the difference of two MCFs and confidence intervals. This is useful for comparing the repair performance of two systems.

Refer to Nelson (2002), Nelson (1995), Nelson (1988), Doganaksoy and Nelson (1991), and Nelson and Doganaksoy (1989) for discussions and examples of analysis of recurrence data.

Recurrence Data With Exact Ages

See "Analysis of Recurrence Data on Repairs" on page 1118 and "Comparison of Two Samples of Repair Data" on page 1122 for examples of the analysis of recurrence data with exact ages.

Formulas for the MCF estimator $\hat{M}(t)$ and the variance of the estimator $\text{Var}(\hat{M}(t))$ are given in Nelson (1995). Table 37.52 shows a set of artificial repair data from Nelson (1988). For each system, the data consist of the system and cost for each repair. If you want to compute the MCF for the number of repairs, rather than cost of repairs, then you should set the cost equal to 1 for each repair. A plus sign (+) in place of a cost indicates that the age is a censoring time. The repair history of each system ends with a censoring time.

Table 37.52. System Repair Histories for Artificial Data

Unit	(Age in Months, Cost in $100)			
6	(5,$3)	(12,$1)	(12,+)	
5	(16,+)			
4	(2,$1)	(8,$1)	(16,$2)	(20,+)
3	(18,$3)	(29,+)		
2	(8,$2)	(14,$1)	(26,$1)	(33,+)
1	(19,$2)	(39,$2)	(42,+)	

Table 37.53 illustrates the calculation of the MCF estimate from the data in Table 37.52. The RELIABILITY procedure uses the following rules for computing the MCF estimates.

1. Order all events (repairs and censoring) by age from smallest to largest.
   - If the event ages of the same or different systems are equal, the corresponding data are sorted from the largest repair cost to the smallest. Censoring events always sort as smaller than repair events with equal ages.
   - When event ages and values of more than one system coincide, the corresponding data are sorted from the largest system identifier to the smallest. The system IDs can be numeric or character, but they are always sorted in ASCII order.
2. Compute the number of systems I in service at the current age as the number in service at the last repair time minus the number of censored units in the intervening times.
3. For each repair, compute the mean cost as the cost of the current repair divided by the number in service I.
4. Compute the MCF for each repair as the previous MCF plus the mean cost for the current repair.

Table 37.53. Calculation of MCF for Artificial Data

Event	(Age,Cost)	Number I in Service	Mean Cost	MCF
1	(2,$1)	6	$1/6=0.17	0.17
2	(5,$3)	6	$3/6=0.50	0.67
3	(8,$2)	6	$2/6=0.33	1.00
4	(8,$1)	6	$1/6=0.17	1.17
5	(12,$1)	6	$1/6=0.17	1.33
6	(12,+)	5		
7	(14,$1)	5	$1/5=0.20	1.53
8	(16,$2)	5	$2/5=0.40	1.93
9	(16,+)	4		
10	(18,$3)	4	$3/4=0.75	2.68
11	(19,$2)	4	$2/4=0.50	3.18
12	(20,+)	3		
13	(26,$1)	3	$1/3=0.33	3.52
14	(29,+)	2		
15	(33,+)	1		
16	(39,$2)	1	$2/1=2.00	5.52
17	(42,+)	0		

The variance of the estimator of the MCF Var$(\hat{M}(t))$ is computed as in Nelson (1995). If the VARIANCE=LAWLESS or VARMETHOD2 option is specified, the

method of Lawless and Nadeau (1995) is used to compute the variance of the estimator of the MCF. This method is recommended if the number of systems or events is large or if a FREQ statement is used to specify a frequency variable.

Default approximate two-sided $\gamma \times 100\%$ pointwise confidence limits for $M(t)$ are computed as

$$M_L(t) = \hat{M}(t) - K_\gamma \sqrt{\mathrm{Var}(\hat{M}(t))}$$

$$M_U(t) = \hat{M}(t) + K_\gamma \sqrt{\mathrm{Var}(\hat{M}(t))}$$

where K_γ represents the $100(1+\gamma)/2$ percentile of the standard normal distribution.

If you specify the LOGINTERVALS option in the MCFPLOT statement, alternative confidence intervals based on the asymptotic normality of $\log(\hat{M}(t))$, rather than of $\hat{M}(t)$, are computed. Let

$$w = \exp\left[\frac{K_\gamma \sqrt{\mathrm{Var}(\hat{M}(t))}}{\hat{M}(t)}\right]$$

Then the limits are computed as

$$M_L(t) = \frac{\hat{M}(t)}{w}$$

$$M_U(t) = \hat{M}(t) \times w$$

These alternative limits are always positive, and can provide better coverage than the default limits when the MCF is known to be positive, such as for counts, or for positive costs. They are not appropriate for MCF differences, and are not computed in this case.

The following SAS statements create the tabular output shown in Figure 37.46 and the plot shown in Figure 37.47.

```
data Art;
   input Sysid$ Time Cost;
   cards;
   sys1 19 2
   sys1 39 2
   sys1 42 -1
   sys2 8  2
   sys2 14 1
   sys2 26 1
   sys2 33 -1
   sys3 18 3
   sys3 29 -1
   sys4 16 2
   sys4 2  1
   sys4 20  -1
   sys4 8  1
```

```
      sys5 16 -1
      sys6 5 3
      sys6 12 1
      sys6 12 -1
      ;
   run;

proc reliability data=Art;
   unitid Sysid;
   mcfplot Time*Cost(-1) ;
run;
```

The first table in Figure 37.46 displays the input data set, the number of observations used in the analysis, the number of systems (units), and the number of repair events. The second table displays the system age, MCF estimate, standard error, approximate confidence limits, and system ID for each event.

The RELIABILITY Procedure

Recurrence Data Summary

Input Data Set	WORK.ART
Observations Used	17
Number of Units	6
Number of Events	11

Recurrence Data Analysis

Age	Sample MCF	Standard Error	95% Confidence Limits Lower	Upper	Unit ID
2.00	0.167	0.167	-0.160	0.493	sys4
5.00	0.667	0.494	-0.302	1.636	sys6
8.00	1.000	0.516	-0.012	2.012	sys2
8.00	1.167	0.543	0.103	2.230	sys4
12.00	1.333	0.667	0.027	2.640	sys6
12.00	.	.	.	.	sys6
14.00	1.533	0.764	0.035	3.032	sys2
16.00	1.933	0.951	0.069	3.797	sys4
16.00	.	.	.	.	sys5
18.00	2.683	0.913	0.894	4.473	sys3
19.00	3.183	0.641	1.926	4.440	sys1
20.00	.	.	.	.	sys4
26.00	3.517	0.679	2.185	4.848	sys2
29.00	.	.	.	.	sys3
33.00	.	.	.	.	sys2
39.00	5.517	0.679	4.185	6.848	sys1
42.00	.	.	.	.	sys1

Figure 37.46. PROC RELIABILITY Output for the Artificial Data

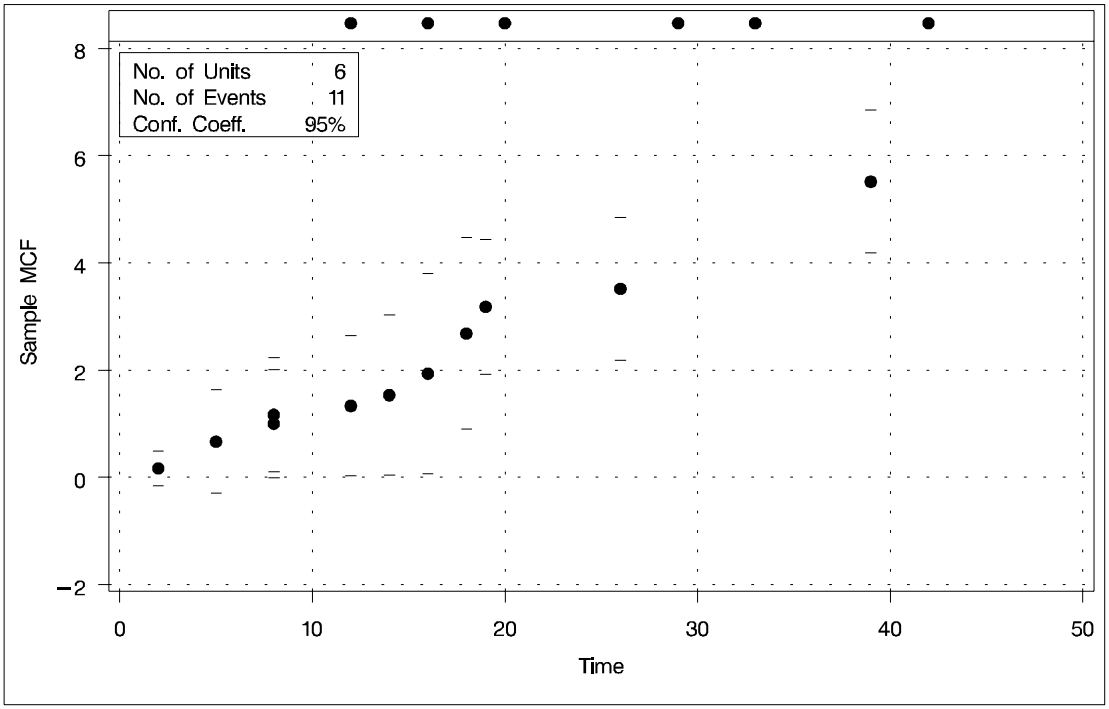


Figure 37.47. MCF Plot for the Artificial Data

Estimates of the difference between two MCFs MDIFF$(t) = M_1(t) - M_2(t)$ and the variance of the estimator are computed as in Doganaksoy and Nelson (1991). Confidence limits for the MCF difference function are computed in the same way as for the MCF, using the variance of the MCF difference function estimator.

Recurrence Data with Ages Grouped into Intervals

Recurrence data are sometimes grouped into time intervals for convenience, or to reduce the number of data records to be stored and analyzed. Interval recurrence data consists of the number of recurrences and the number of censored units in each time interval.

You can use PROC RELIABILITY to compute and plot MCFs and MCF differences for interval data. Formulas for the MCF estimator $\hat{M}(t)$ and the variance of the estimator Var$(\hat{M}(t))$ for interval data, as well as examples and interpretations, are given in Nelson (2002, Chapter 5). These calculations apply only to the number of recurrences, and not to cost.

Let N_0 be the total number of units, R_i the number of recurrences in interval i, $i = 1, \ldots, n$, and C_i the number of units censored into interval i. Then $N_0 = \sum_{i=1}^{n} C_i$ and the number entering interval i is $N_i = N_{i-1} - C_{i-1}$ with $C_0 = 0$. The MCF estimate for interval i is $M_0 = 0$,

$$M_i = M_{i-1} + \frac{R_i}{N_i - .5C_i}$$

The denominator $N_i - .5C_i$ approximates the number at risk in interval i, and treats the censored units as if they were censored halfway through the interval. Since no censored units are likely to have ages lasting through the entire last interval, the MCF estimate for the last interval is likely to be biased. A footnote is printed in the tabular output to as a reminder of this bias for the last interval.

See "Analysis of Interval Age Recurrence Data" on page 1126 for an example of interval recurrence data analysis.

ODS Table Names

The following tables contain the ODS table names created by the RELIABILITY Procedure, organized by the statements that produce them.

Table 37.54. Tables Produced with the ANALYZE Statement

Table Name	Description
ConvergenceStatus	convergence status
CorrMat	parameter correlation matrix
CovMat	parameter covariance matrix
DatSum	summary of fit
GradHess	last evaluation of parameters, gradient, and Hessian
IterEM	iteration history for Turnbull algorithm
IterLRParm	iteration history for likelihood ratio confidence intervals for parameters
IterLRPer	iteration history for likelihood ratio confidence intervals for percentiles
IterParms	iteration history for parameter estimates
Lagrange	Lagrange multiplier statistics
NObs	Observations Summary
PBEst	Poisson/binomial estimates by group
PBPred	Poisson/binomial predicted values
PBPredTol	Poisson/binomial predicted values by group
PBSum	Poisson/binomial analysis summary
PBTol	Poisson/binomial tolerance estimates
PctEst	percentile estimates
ParmEst	parameter estimates
ParmOther	fitted distribution mean, median, mode
PGradHess	last evaluation of parameters, gradient, and Hessian in terms of stable parameters
ProbabilityEstimates	nonparametric cumulative distribution function estimates
RelInfo	model information
SurvEst	survival function estimates
TurnbullGrad	interval probabilities, reduced gradient, Lagrange multipliers for Turnbull algorithm
WCorrMat	parameter correlation matrix for Weibull distribution
WCovMat	parameter covariance matrix for Weibull distribution

Table 37.55. Tables Produced with the MCFPLOT Statement

Table Name	Description
McfDEst	MCF difference estimates
McfDSum	MCF difference data summary
McfEst	MCF estimates
McfSum	MCF data summary

Table 37.56. Tables Produced with the MODEL Statement

Table Name	Description
MConvergenceStatus	convergence status
ClassLevels	class level information
ModCorMat	parameter correlation matrix
ModCovMat	parameter covariance matrix
ModFitSum	summary of fit
ModInfo	model information
ModIterLRparm	iteration history for likelihood ratio confidence intervals for parameters
ModIterParms	iteration history for parameter estimates
ModLagr	Lagrange multiplier statistics
ModLastGradHess	last evaluation of the gradient and Hessian
ModNObs	Observations Summary
ModObstats	observation statistics
ModParmInfo	parameter information
ModPrmEst	parameter estimates

Table 37.57. Tables Produced with PROBPLOT and RELATIONPLOT Statements

Table Name	Description
ConvergenceStatus	convergence status
CorrMat	parameter correlation matrix
CovMat	parameter covariance matrix
DatSum	summary of fit
GradHess	last evaluation of parameters, gradient, and Hessian
IterEM	iteration history for Turnbull algorithm
IterLRParm	iteration history for likelihood ratio confidence intervals for parameters
IterLRPer	iteration history for likelihood ratio confidence intervals for percentiles
IterParms	iteration history for parameter estimates
Lagrange	Lagrange multiplier statistics
NObs	Observations Summary
PctEst	percentile estimates
ParmEst	parameter estimates
ParmOther	fitted distribution mean, median, mode
PGradHess	last evaluation of parameters, gradient, and Hessian in terms of stable parameters
ProbabilityEstimates	nonparametric cumulative distribution function estimates
RelInfo	model information
SurvEst	survival function estimates

Table 37.57. Tables Produced with PROBPLOT and RELATIONPLOT Statements (continued)

Table Name	Description
TurnbullGrad	interval probabilities, reduced gradient, Lagrange multipliers for Turnbull algorithm
WCorrMat	parameter correlation matrix for Weibull distribution
WCovMat	parameter covariance matrix for Weibull distribution

References

Abernethy, Robert B. (1996) (Author and Publisher), *The New Weibull Handbook*, 536 Oyster Road, North Palm Beach, FL 33408-4328.

Ascher, H. and Feingold, H. (1984), *Repairable Systems Reliability*, New York: Marcel Dekker, Inc.

Collett, D. (1994), *Modelling Survival Data In Medical Research*, London: Chapman and Hall.

Doganaksoy, N. and Nelson, W. (1991), "A Method and Computer Program MCFDIFF to Compare Two Samples of Repair Data," GE Research & Development Center TIS Report 91CRD172, P.O. Box 8, Schenectady, NY 12301.

Doganaksoy, N. and Schmee, J. (1993), "Orthogonal Parameters with Censored Data," *Commun. Statist. - Theory Meth.*, 22 (3), 669–685.

Doganaksoy, N., Hahn, G.J., and Meeker, W.G. (2002), "Reliability Analysis by Failure Mode," *Quality Progress*, 35 (6), 47–52.

Gentleman, R. and Geyer, C.J. (1994), "Maximum Likelihood for Interval Censored Data: Consistency and Computation," *Biometrika*, 81 (3), 618–623.

Joe, H. and Proschan, F. (1984), "Percentile Residual Life Functions," *Operations Research*, 32 (3), 668–678.

Johnson, N.L., Kotz, S., and Kemp, A.W. (1992), *Univariate Discrete Distributions*, Second Edition, New York: John Wiley & Sons.

Lawless, J.F. (1982), *Statistical Models and Methods for Lifetime Data*, New York: John Wiley & Sons.

Lawless, J.F. and Nadeau, C. (1995), "Some Simple Robust Methods for the Analysis of Recurrent Events," *Technometrics*, 37, 158–168.

Meeker, W.Q. and Escobar, L.A. (1998), *Statistical Methods for Reliability Data*, New York: John Wiley & Sons.

Nair, V.N. (1984), "Confidence Bands for Survival Functions with Censored Data: A Comparative Study," *Technometrics*, 26 (3), 265–275.

Nelson, W. (1982), *Applied Life Data Analysis*, New York: John Wiley & Sons.

Nelson, W. (1985), "Weibull Analysis of Reliability Data with Few or No Failures," *Journal of Quality Technology*, 17 (3), 140–146.

Nelson, W. (1988), "Graphical Analysis of System Repair Data," *Journal of Quality Technology*, 20 (1), 24–35.

Nelson, W. (1990), *Accelerated Testing: Statistical Models, Test Plans, and Data Analyses*, New York: John Wiley & Sons.

Nelson, W. (1995), "Confidence Limits for Recurrence Data—Applied to Cost or Number of Product Repairs," *Technometrics*, 37, 147–157.

Nelson, Wayne B. (2002), *Recurrent Events Data Analysis for Product Repairs, Disease Recurrences, and Other Applications*, ASA-SIAM Series on Statistics and Applied Probability, SIAM, Philadelphia, ASA, Alexandria, VA.

Nelson, W. and Doganaksoy, N. (1989), "A Computer Program for an Estimate and Confidence Limits for the Mean Cumulative Function for Cost or Number of Repairs of Repairable Products," GE Research & Development Center TIS Report 89CRD239, P.O. Box 8, Schenectady, NY 12301.

Tobias, P.A. and Trindade, D.C. (1995), *Applied Reliability*, Second Edition, New York: Van Nostrand Reinhold.

Turnbull, B.W. (1976), "The Empirical Distribution Function with Arbitrarily Grouped, Censored and Truncated Data," *J. R. Statist. Soc.* B, 38, 290–295.

Part 10
The SHEWHART Procedure

Contents

Introduction 1221

Chapter 38. PROC SHEWHART and General Statements 1227

Chapter 39. BOXCHART Statement 1237

Chapter 40. CCHART Statement 1303

Chapter 41. IRCHART Statement 1345

Chapter 42. MCHART Statement 1389

Chapter 43. MRCHART Statement 1435

Chapter 44. NPCHART Statement 1481

Chapter 45. PCHART Statement 1525

Chapter 46. RCHART Statement 1571

Chapter 47. SCHART Statement 1611

Chapter 48. UCHART Statement 1649

Chapter 49. XCHART Statement 1689

Chapter 50. XRCHART Statement 1735

Chapter 51. XSCHART Statement 1787

Chapter 52. INSET and INSET2 Statements 1833

Chapter 53. Dictionary of Options 1851

Chapter 54. Graphical Enhancements 1927

Chapter 55. Tests for Special Causes 1975

Chapter 56. Specialized Control Charts 1999

Chapter 57. Interactive Control Charts 2039

References 2049

Introduction

The Shewhart control chart is a graphical and analytical tool for deciding whether a process is in a state of statistical control. You can use the SHEWHART procedure to display many different types of control charts, including all commonly used charts for variables and attributes. In addition, you can use the SHEWHART procedure to

- create charts from either raw data (actual measurements) or summarized data
- analyze multiple process variables
- specify control limits in terms of a multiple of the standard error of the plotted summary statistic or as probability limits
- adjust control limits to compensate for unequal subgroup sizes
- estimate control limits from the data, compute control limits from specified values for population parameters (known standards), or read limits from an input data set
- create historical control charts that display distinct sets of control limits for multiple time phases
- perform tests for special causes based on runs patterns (Western Electric rules)
- estimate the process standard deviation using various methods (variable charts only)
- accept numeric-valued or character-valued subgroup variables
- display subgroups with date and time formats
- save chart statistics and control limits in output data sets
- tabulate chart statistics and control limits
- generate charts on line printers or on graphics devices. Charts produced on line printers can use special formatting characters that improve the appearance of the chart. Charts produced on graphics devices can be annotated, saved, and replayed.

Uses of Shewhart Charts

The Shewhart chart is named after Walter A. Shewhart (1891-1967), a physicist at the Bell Telephone Laboratories, who introduced the method in 1924 and elaborated upon it in his book *Economic Control of Quality of Manufactured Product*, (1931). The concepts underlying the control chart are that the natural variability in any manufacturing process can be quantified with a set of control limits and that the variation exceeding these limits signals a change in the process.

In industry, the Shewhart chart is the most commonly applied statistical quality control method for studying the variation in output from a manufacturing process.

Shewhart charts are typically used to distinguish variation due to *special causes* from variation due to *common causes*. Special causes, also referred to as *assignable causes*, are local, sporadic problems such as the failure of a particular machine or a mistakenly recorded measurement. Common causes are problems inherent in the manufacturing system as a whole. Examples of common causes are inadequate product design, inherited defective material, and excessive humidity.

When the special causes have been identified and eliminated, the process is said to be in *statistical control*. Once statistical control has been established, Shewhart charts can be used to monitor the process for the occurrence of future special causes and to measure and reduce the effects of common causes.

Deming (1982) emphasized that the improvement of a process can begin only after statistical control has been established. Deming also noted that control chart techniques are applicable to quality improvement in service industries as well as manufacturing industries.

Characteristics of Shewhart Charts

Figure 1 illustrates a typical Shewhart chart.

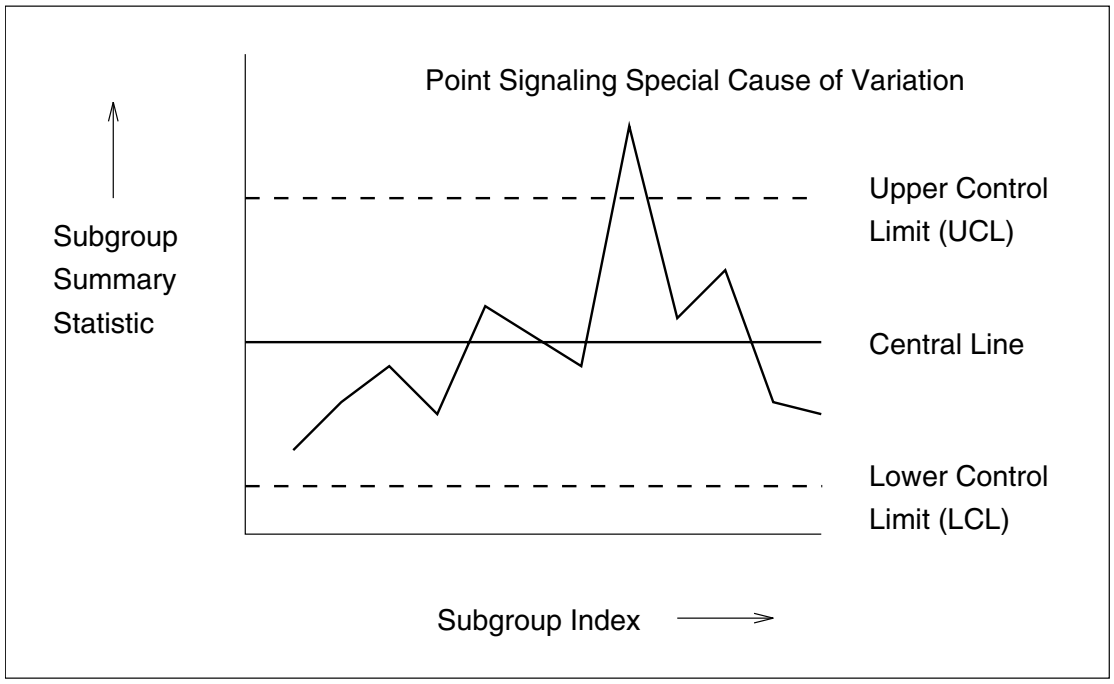


Figure 1. A Shewhart Control Chart

All Shewhart charts have the following characteristics:

- Each point represents a *summary statistic* computed from a sample of measurements of a quality characteristic. For example, the summary statistic might be the average value of a critical dimension of five items selected at random, or it might be the proportion of nonconforming items in a sample of 100 items.
- The *vertical axis* of a Shewhart chart is scaled in the same units as the summary statistic.
- The samples from which the summary statistics are computed are referred to as *rational subgroups* or *subgroup samples*. The organization of the data into subgroups is critical to the interpretation of a Shewhart chart. Shewhart (1931) advocated selecting rational subgroups so that variation within subgroups is minimized and variation among subgroups is maximized; this makes the chart more sensitive to shifts in the process level. Various approaches to subgrouping are discussed by Grant and Leavenworth (1980), Montgomery (1996), and Kume (1985).
- The *horizontal axis* of a Shewhart chart identifies the subgroup samples. Frequently, the samples are indexed according to the order in which they were taken or the time at which they were taken. Subgroup samples can also be assigned labels that indicate some other type of classification (for example, lot number).
- The *central line* on a Shewhart chart indicates the average (expected value) of the summary statistic when the process is in statistical control.
- The *upper and lower control limits*, labeled UCL and LCL, respectively, indicate the range of variation to be expected in the summary statistic when the process is in statistical control. The control limits are commonly computed as 3σ limits* representing three standard errors† of variation in the summary statistic above and below the central line. However, the limits can also be determined using a multiple of the standard error other than three, or from a specified probability (α) that a single summary statistic will exceed the limits when the process is in statistical control. Limits determined by the latter method are referred to as *probability limits*.

  The control limits are also determined by the subgroup sample size because the standard error of the summary statistic is a function of sample size. If the sample size is constant across subgroups, the control limits are typically horizontal lines, as in Figure 1. However, if the sample size varies from subgroup to subgroup, the limits are usually adjusted to compensate for the effect of sample size, resulting in step-like boundaries.

  Control limits can be estimated from the data being analyzed, or they can be standard, previously determined values. Estimated limits are often used when

*In this context, the symbol σ always stands for the standard error of the subgroup summary statistic that is plotted on the chart. Elsewhere in this chapter, σ is also used to denote the standard deviation of a process, also referred to as the population standard deviation. This dual usage is standard practice.

†The term *standard deviation* is also used by some authors to refer to this quantity; see, for example, Montgomery (1996). This chapter uses the term *standard error* for the dispersion of the distribution of a statistic and the term *standard deviation* for the dispersion of a distribution of individual measurements.

statistical control is being established, and standard limits are often used when statistical control is being maintained.

- *A point outside the control limits* signals the presence of a special cause of variation. Additionally, *tests for special causes* (also referred to as *Western Electric rules* and *runs tests*) can signal an out-of-control condition if a statistically unusual pattern of points is observed in the control chart. For example, one pattern used to diagnose the existence of a trend is seven consecutive steadily increasing points.

  When the process is in statistical control, a point may fall outside the control limits purely by chance, resulting in a false out-of-control signal. However, when the Shewhart chart correctly signals the presence of a special cause, additional action is needed to determine the nature of the problem and eliminate it.

Classification of Shewhart Charts

Shewhart charts are broadly classified according to the type of data analyzed.

- Shewhart charts for *variables* are used when the quality characteristic of a process is measured on a continuous scale.
- Shewhart charts for *attributes* are used when the quality characteristic of a process is measured by counting the number of nonconformities (defects) in an item or the number of nonconforming (defective) items in a sample.

Shewhart charts for variables are further classified according to the subgroup summary statistic plotted on the chart.

- $\bar{X}$ and R charts display subgroup means (averages) and ranges. Typically the two charts are presented on the same page, with the $\bar{X}$ chart aligned above the R chart to facilitate the simultaneous analysis of the central tendency and variability of the process.
- $\bar{X}$ and s charts display subgroup means (averages) and standard deviations. Typically the two charts are presented on the same page, with the $\bar{X}$ chart aligned above the s chart.
- Median and range charts display subgroup medians and ranges. Typically the two charts are presented on the same page, with the median chart aligned above the R chart.
- Charts for individual measurements and moving ranges display individual measurements and moving ranges of two of more successive measurements. In this case the subgroup sample consists of a single observation.

Likewise, Shewhart charts for attributes are classified according to the subgroup summary statistic plotted on the chart:

- A p chart displays the proportion of nonconforming (defective) items in a subgroup sample.
- An np chart displays the number of nonconforming (defective) items in a subgroup sample.
- A u chart displays the number of nonconformities (defects) per unit in a subgroup sample consisting of an arbitrary number of units.
- A c chart displays the number of nonconformities (defects) in a unit (here, a subgroup sample typically consists of one unit).

You can create all of the preceding types of Shewhart charts with the SHEWHART procedure. In addition, you can create a wide variety of nonstandard Shewhart charts, including

- a trend chart displaying a time trend plot and an $\bar{X}$ chart (or median chart) that has been created removing the time trend from the data. The trend chart and $\bar{X}$ chart are presented on the same page, with the $\bar{X}$ aligned above the trend chart, to facilitate the detection of special causes after accounting for the time trend effect. Trend charts are applicable when a time trend (for instance, due to tool wear) is observed in a preliminary $\bar{X}$ chart of the original data.
- a box chart displaying a box plot (box-and-whisker plot) for each subgroup and control limits for the subgroup means. This chart facilitates detailed analysis of the subgroup distributions and is applicable with large subgroup sample sizes (ten or more).

Learning to Use the SHEWHART Procedure

Although the SHEWHART procedure provides a large number of options, you can use the procedure to create a basic Shewhart chart with as few as two SAS statements:

- the PROC SHEWHART statement, which starts the procedure and specifies the input SAS data set
- a chart statement, which specifies the type of Shewhart chart you want to create and the variables in the input data set that you want to analyze

For example, you can use the following statements to create $\bar{X}$ and R charts with 3σ limits for measurements read from a SAS data set named DRUMS:

```
proc shewhart data=drums graphics;
   xrchart flwidth*hour;
run;
```

The keyword XRCHART in the chart statement specifies that $\bar{X}$ and R charts are to be created. The following SAS variables are specified in the XRCHART statement:

- A SAS variable (FLWDITH), whose values are the process measurements, is specified before the asterisk. This variable is referred to as the *process*.
- A SAS variable (HOUR), whose values classify the measurements into subgroups, is specified after the asterisk. This variable is referred to as a *subgroup-variable*.

The same form of specification is used with other chart statements to create different types of Shewhart charts. The following table lists the keywords for the 13 chart statements that are available with the SHEWHART procedure:

Table 1. Chart Statements in the SHEWHART Procedure

Keyword	Chart(s) Displayed	"Getting Started" Page
BOXCHART	box chart with optional trend chart	1240
CCHART	c chart	1306
IRCHART	individual and moving range charts	1348
MCHART	median chart with optional trend chart	1392
MRCHART	median and R charts	1438
NPCHART	np chart	1484
PCHART	p chart	1528
RCHART	R chart	1574
SCHART	s chart	1614
UCHART	u chart	1652
XCHART	$\bar{X}$ chart with optional trend chart	1692
XRCHART	$\bar{X}$ and R charts	1738
XSCHART	$\bar{X}$ and s charts	1790

If you are using the SHEWHART procedure for the first time, you should

- read Chapter 38, "PROC SHEWHART and General Statements."
- read the "Getting Started" section in the chapter for the chart statement you need to create your chart. Table 1 lists the pages for these sections.

Once you have learned to use a particular chart statement, you will find it straightforward to use the remaining chart statements since their syntax is nearly the same. A separate, self-contained chapter is provided for each chart statement.

Chapter 38
PROC SHEWHART and General Statements

Chapter Contents

OVERVIEW 1229

SYNTAX OVERVIEW FOR THE SHEWHART PROCEDURE 1230
- BY and ID Statements 1231
- Graphical Enhancement Statements 1231

SYNTAX FOR THE PROC SHEWHART STATEMENT 1232
- Summary of Options 1232
- Dictionary of Options 1232

INPUT AND OUTPUT DATA SETS 1236

Chapter 38
PROC SHEWHART and General Statements

Overview

The PROC SHEWHART statement starts the SHEWHART procedure and it optionally identifies various data sets.

To create a Shewhart chart, you specify a chart statement (after the PROC SHEWHART statement) that specifies the type of Shewhart chart you want to create and the variables in the input data set that you want to analyze. For example, the following statements request $\bar{X}$ and R charts:

```
proc shewhart data=values;
   xrchart weight*lot;
run;
```

Here, the DATA= option specifies an input data set (VALUES) with the *process* measurement variable (WEIGHT) and the *subgroup-variable* (LOT). *

You can use options in the PROC SHEWHART statement to

- specify input data sets containing variables to be analyzed, control limit information, or annotation information
- specify a graphics catalog for saving graphical output
- specify whether charts are to be produced on graphics devices or line printers
- define characters used for features on charts produced on line printers

Note: If you are learning to use the SHEWHART procedure, you should read both this chapter and the "Getting Started" section in the chapter for the chart statement that corresponds to the chart you want to create.

*In Release 6.12 and previous releases of SAS/QC software, the keyword GRAPHICS was required in the PROC SHEWHART statement to specify that the chart be created with a graphics device. In Release 7 and later releases, you can specify the LINEPRINTER option to request line printer plots.

Syntax Overview for the SHEWHART Procedure

The following are the primary statements that control the SHEWHART procedure:

PROC SHEWHART < *options* >;

BOXCHART (*processes*)***subgroup-variable* <(*block-variables*)>
< =*symbol-variable* | ='*character*' > < **/** *options* >;

CCHART (*processes*)***subgroup-variable* <(*block-variables*)>
< =*symbol-variable* | ='*character*' > < **/** *options* >;

IRCHART (*processes*)***subgroup-variable* <(*block-variables*)>
< =*symbol-variable* | ='*character*' > < **/** *options* >;

MCHART (*processes*)***subgroup-variable* <(*block-variables*)>
< =*symbol-variable* | ='*character*' > < **/** *options* >;

MRCHART (*processes*)***subgroup-variable* <(*block-variables*)>
< =*symbol-variable* | ='*character*' > < **/** *options* >;

NPCHART (*processes*)***subgroup-variable* <(*block-variables*)>
< =*symbol-variable* | ='*character*' > < **/** *options* >;

PCHART (*processes*)***subgroup-variable* <(*block-variables*)>
< =*symbol-variable* | ='*character*' > < **/** *options* >;

RCHART (*processes*)***subgroup-variable* <(*block-variables*)>
< =*symbol-variable* | ='*character*' > < **/** *options* >;

SCHART (*processes*)***subgroup-variable* <(*block-variables*)>
< =*symbol-variable* | ='*character*' > < **/** *options* >;

UCHART (*processes*)***subgroup-variable* <(*block-variables*)>
< =*symbol-variable* | ='*character*' > < **/** *options* >;

XCHART (*processes*)***subgroup-variable* <(*block-variables*)>
< =*symbol-variable* | ='*character*' > < **/** *options* >;

XRCHART (*processes*)***subgroup-variable* <(*block-variables*)>
< =*symbol-variable* | ='*character*' > < **/** *options* >;

XSCHART (*processes*)***subgroup-variable* <(*block-variables*)>
< =*symbol-variable* | ='*character*' > < **/** *options* >;

INSET *keyword-list* < **/** *options* >;

INSET2 *keyword-list* < **/** *options* >;

The PROC SHEWHART statement invokes the procedure and specifies the input data set. The chart statements create different types of control charts. You can specify one or more of each of the chart statements. For details, read the chapter on the chart statement that corresponds to the type of control chart you want to produce.

BY and ID Statements

In addition, you can optionally specify one of each of the following statements:

BY *variables*;
ID *variables*;

The BY statement specifies variables in the input data set that are used for BY processing. A separate control chart is created for each group of observations defined by the levels of the BY variables. The input data set must be sorted in order of the BY variables.

The ID statement specifies variables used to identify observations. The ID variables must be variables in the DATA= or HISTORY= input data sets.

The ID variables are used in the following ways:

- If you create an OUTHISTORY= or OUTTABLE= data set, the ID variables are included. If the input data set is a DATA= data set, only the values of the ID variables from the first observation in each subgroup are passed to the output data set.
- If you specify the TABLEID or TABLEALL options in a chart statement, the table produced is augmented by a column for each of the ID variables. Only the values of the ID variables from the first observation in each subgroup are tabulated. See the entry for the TABLEID option (page 1911) in Chapter 53, "Dictionary of Options."
- If you specify the BOXSTYLE=SCHEMATICID option or the BOXSTYLE= SCHEMATICIDFAR option in the BOXCHART statement, the value of the first variable listed in the ID statement is used to label each extreme observation. See Output 39.2.3 on page 1289 and Output 39.2.4 on page 1290.

Graphical Enhancement Statements

You can use TITLE, FOOTNOTE, and NOTE statements to enhance graphical and printed output. If you are creating charts with a graphics device, you can also use AXIS, LEGEND, and SYMBOL statements to enhance your charts. For details, refer to *SAS/GRAPH Software: Reference* and see the chapter for the control chart statement that you are using.

Syntax for the PROC SHEWHART Statement

The syntax for the PROC SHEWHART statement is as follows:

PROC SHEWHART < *options* >**;**

The PROC SHEWHART statement starts the SHEWHART procedure, and it optionally identifies various data sets and requests graphics output. The following section lists all *options*. See "Dictionary of Options" below for detailed information.

Summary of Options

The following tables list the PROC SHEWHART *options* by function:

Table 38.1. Input Data Sets Options

ANNOTATE=*SAS-data-set*	specifies input data set containing annotation information for primary chart
ANNOTATE2=*SAS-data-set*	specifies input data set containing annotation information for secondary chart
BOX=*SAS-data-set*	specifies input data set containing summary statistics, control limits, and box chart outlier values
DATA=*SAS-data-set*	specifies input data set containing raw data
HISTORY=*SAS-data-set*	specifies input data set containing summary statistics
LIMITS=*SAS-data-set*	specifies input data set containing control limits
TABLE=*SAS-data-set*	specifies input data set containing summary statistics and control limits
TESTURLS=*SAS-data-set*	specifies input data set containing URLs associated with subgroups with positive tests for special causes

Table 38.2. Plotting and Graphics Options

FORMCHAR(*index*)='*string*'	defines characters used for features on charts
GOUT=*graphics-catalog*	specifies catalog for saving graphical output
LINEPRINTER	requests line printer charts be produced

Dictionary of Options

The following entries provide detailed descriptions of options in the PROC SHEWHART statement. The marginal notes *Graphics* and *Line Printer* identify options that apply to graphics devices and line printers, respectively.

ANNOTATE=*SAS-data-set*
ANNO=*SAS-data-set*

Graphics

specifies an input data set containing Annotate variables as described in *SAS/GRAPH Software: Reference*. You can use this data set to add features to primary charts produced on graphics devices; use this data set only when the chart is created using a graphics device; it is ignored when then LINEPRINTER option is specified. Features provided in this data set are displayed on every chart produced in the current run of PROC SHEWHART.

ANNOTATE2=*SAS-data-set*
ANNO2=*SAS-data-set*

specifies an input data set that contains annotate variables. You can use this data set to add features to the secondary chart in statements that produce two charts (the IRCHART, MRCHART, XRCHART, and XSCHART statements and, when you specify the TRENDVAR= option, the BOXCHART, MCHART, and XCHART statements). The restrictions and features are the same as those for the ANNOTATE= option. *Graphics*

BOX=*SAS-data-set*

names an input data set that contains subgroup summary statistics, control limits, and outlier values in "strung out" form, with more than one observation per subgroup. Each observation corresponds to one feature of one subgroup's box-and-whisker plot. Typically, this data set is created as an OUTBOX= data set in a previous run of PROC SHEWHART with a BOXCHART statement. The BOX= data set is the only kind of summary data set you can use to produce schematic box-and-whisker plots. The BOXCHART statement is the only chart statement you can use with a BOX= input data set.

DATA=*SAS-data-set*

names an input data set that contains raw data as observations. Note that the DATA= data set may need sorting. If the values of the *subgroup-variable* are numeric, you must sort the data set so that these values are in increasing order (within BY groups). Use PROC SORT if the data are not already sorted.

The DATA= data set may contain more than one observation for each value of the *subgroup-variable*. This happens, for example, when you produce a control chart for means and ranges with the XRCHART statement.

You cannot use a DATA= data set together with a HISTORY= or a TABLE= data set. If you do not specify one of these three input data sets, PROC SHEWHART uses the most recently created data set as a DATA= data set. For more information, see the "DATA= Data Set" section in the chapter for the chart statement you are using.

FORMCHAR(*index***)='***string***'**

defines characters used for features on charts produced on a line printer, where *index* is a list of numbers ranging from 1 to 17, and *string* is a character or hexadecimal string. The *index* identifies which features are controlled with the *string* characters, as discussed in the following table. If you specify the FORMCHAR= option and omit the *index*, the *string* controls all 17 features. *Line Printer*

Value of *index*	Description of Character	Chart Feature
1	vertical bar	frame
2	horizontal bar	frame, central line
3	box character (upper left)	frame
4	box character (upper middle)	serifs, tick (horizontal axis)
5	box character (upper right)	frame
6	box character (middle left)	not used
7	box character (middle middle)	serifs
8	box character (middle right)	tick (vertical axis)
9	box character (lower left)	frame
10	box character (lower middle)	serifs
11	box character (lower right)	frame
12	vertical bar	control limits
13	horizontal bar	control limits
14	box character (upper right)	control limits
15	box character (lower left)	control limits
16	box character (lower right)	control limits
17	box character (upper left)	control limits

Not all printers can produce all the characters in the preceding list. By default, the form character list specified with the SAS system FORMCHAR= option is used; otherwise, the default is FORMCHAR='|—-|+|—|====='. If you print to a PC screen or if your device supports the ASCII symbol set (1 or 2), the following is recommended:

```
formchar='B3,C4,DA,C2,BF,C3,C5,B4,C0,C1,D9,BA,CD,BB,C8,BC,D9'X
```

Note that the FORMCHAR= option in the PROC SHEWHART statement enables you to override temporarily the values of the SAS system option of the same name. The values of the SAS system option are not altered by using the FORMCHAR= option in the PROC SHEWHART statement.

GOUT=*graphics-catalog*

Graphics

specifies the graphics catalog for graphics output from PROC SHEWHART. This is useful if you want to save the output. The GOUT= option is used only when the chart is created using a graphics device; it is ignored when the LINEPRINTER option is specified.

HISTORY=*SAS-data-set*

HIST=*SAS-data-set*

names an input data set that contains subgroup summary statistics. For example, you can read sample sizes, means, and ranges for the subgroups to create $\bar{X}$ and R charts. Typically, this data set is created as an OUTHISTORY= data set in a previous run of PROC SHEWHART, but it can also be created using a SAS summarization procedure such as PROC MEANS.

Note that the HISTORY= data sets may need sorting. If the values of the *subgroup-variable* are numeric, you need to sort the data set so that these values are in increasing order (within BY groups). Use PROC SORT if the data are not already sorted.

The HISTORY= data set can contain only one observation for each value for the *subgroup-variable*.

You cannot use a HISTORY= data set with a DATA= or a TABLE= data set. If you do not specify one of these three input data sets, PROC SHEWHART uses the most recently created data set as a DATA= data set. For more information, see the "HISTORY= Data Set" section in the chapter for the chart statement you are using.

LIMITS=*SAS-data-set*
names an input data set that contains preestablished control limits or the parameters from which control limits can be computed. Each observation in a LIMITS= data set provides control limit information for a *process*. Typically, this data set is created as an OUTLIMITS= data set in a previous run of PROC SHEWHART.

If you omit the LIMITS= option, then control limits are computed from the data in the DATA= or HISTORY= input data sets. For details about the variables needed in a LIMITS= data set, see the "LIMITS= Data Set" section in the chapter for the chart statement you are using.

LINEPRINTER
requests that line printer charts be produced. By default, the procedure creates charts for a graphics device.

TABLE=*SAS-data-set*
names an input data set that contains subgroup summary statistics and control limits. Each observation in a TABLE= data set provides information for a particular subgroup and *process*. Typically, this data set is created as an OUTTABLE= data set in a previous run of PROC SHEWHART.

You cannot use a TABLE= data set with a DATA= or a HISTORY= data set. If you do not specify one of these three input data sets, PROC SHEWHART uses the most recently created data set as a DATA= data set. For more information, see the "TABLE= Data Set" section in the chapter for the chart statement that you are using.

TESTURLS=*SAS-data-set*
names an input data set that contains variables associated with tests for special causes. A TESTURLS= data set contains variables _TEST_, _CHART_, and _URL_. _TEST_ and _CHART_ are numeric variables identifying a test for special causes (1-8) and the primary or secondary chart (1 or 2). _URL_ is a character variable containing the URL to be associated with subgroups for which the given test on the given chart is positive. See the chapter "Interactive Control Charts" for more information.

Input and Output Data Sets

Figure 38.1 summarizes the input and output data sets used with the SHEWHART procedure.

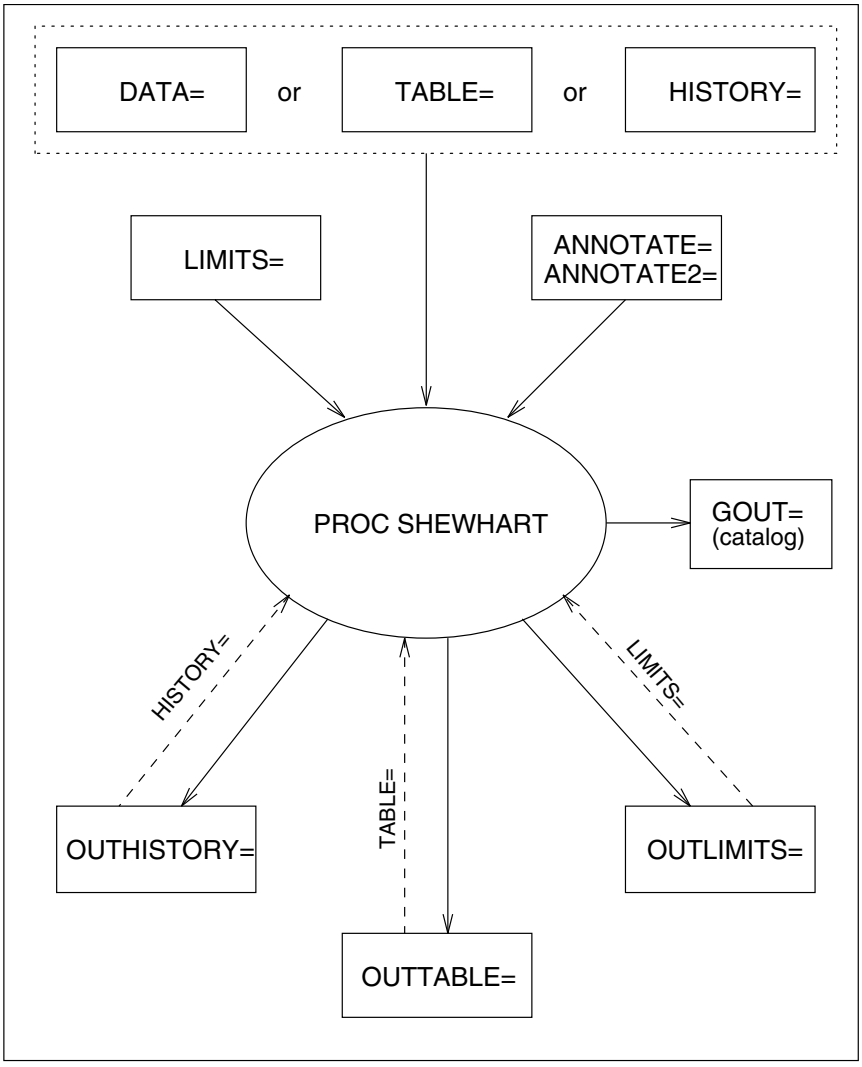


Figure 38.1. Input and Output Data Sets in the SHEWHART Procedure

Chapter 39
BOXCHART Statement

Chapter Contents

OVERVIEW 1239

GETTING STARTED 1240
- Creating Box Charts from Raw Data 1240
- Creating Box Charts from Subgroup Summary Data 1243
- Saving Summary Statistics 1246
- Saving Control Limits 1248
- Reading Preestablished Control Limits 1251

SYNTAX 1252
- Summary of Options 1254

DETAILS 1266
- Constructing Box Charts 1266
- Output Data Sets 1269
- ODS Tables 1274
- Input Data Sets 1274
- Methods for Estimating the Standard Deviation 1280
- Percentile Definitions 1282
- Axis Labels 1283
- Missing Values 1283

EXAMPLES 1284
- Example 39.1. Using Box Charts to Compare Subgroups 1284
- Example 39.2. Creating Various Styles of Box-and-Whisker Plots 1287
- Example 39.3. Creating Notched Box-and-Whisker Plots 1291
- Example 39.4. Creating Box-and-Whisker Plots with Varying Widths 1292
- Example 39.5. Creating Box-and-Whisker Plots with Different Line Styles and Colors 1293
- Example 39.6. Computing the Control Limits for Subgroup Maximums . . . 1295
- Example 39.7. Constructing Multi-Vari Charts 1298

Chapter 39
BOXCHART Statement

Overview

The BOXCHART statement creates an $\bar{X}$ chart for subgroup means superimposed with box-and-whisker plots of the measurements in each subgroup. Throughout this chapter, a chart of this type is referred to as a *box chart*. This chart is recommended for large subgroup sample sizes (typically greater than ten). You can also use the BOXCHART statement to create standard side-by-side box-and-whisker plots (see Example 39.2 on page 1287 and Example 39.3 on page 1291).

You can use options in the BOXCHART statement to

- specify control limits for subgroup means or medians
- compute control limits from the data based on a multiple of the standard error of the means (or medians) or as probability limits
- tabulate subgroup summary statistics and control limits
- save control limits in an output data set
- save subgroup summary statistics in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify one of several methods for estimating the process standard deviation
- specify whether subgroup standard deviations or subgroup ranges are used to estimate the process standard deviation
- specify a known (standard) process mean and standard deviation for computing control limits
- create a secondary chart that displays a time trend removed from the data (see "Displaying Trends in Process Data" on page 1957)
- specify one of several methods for calculating quantile statistics (percentiles)
- control the style of the box-and-whisker plots
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

Getting Started

This section introduces the BOXCHART statement with simple examples that illustrate commonly used options. Complete syntax for the BOXCHART statement is presented in the "Syntax" section on page 1252, and advanced examples are given in the "Examples" section on page 1284.

Creating Box Charts from Raw Data

See SHWBOXA in the SAS/QC Sample Library

A petroleum company uses a turbine to heat water into steam that is pumped into the ground to make oil less viscous and easier to extract. This process occurs 20 times daily, and the amount of power (in kilowatts) used to heat the water to the desired temperature is recorded. The following statements create a SAS data set that contains the power output measurements for 20 days:

```
data turbine;
   informat day date7.;
   format day date5.;
   label kwatts='Average Power Output';
   input day @;
   do i=1 to 10;
      input kwatts @;
      output;
      end;
   drop i;
   datalines;
04JUL94 3196 3507 4050 3215 3583 3617 3789 3180 3505 3454
04JUL94 3417 3199 3613 3384 3475 3316 3556 3607 3364 3721
05JUL94 3390 3562 3413 3193 3635 3179 3348 3199 3413 3562
05JUL94 3428 3320 3745 3426 3849 3256 3841 3575 3752 3347
06JUL94 3478 3465 3445 3383 3684 3304 3398 3578 3348 3369
06JUL94 3670 3614 3307 3595 3448 3304 3385 3499 3781 3711

  ...

23JUL94 3421 3787 3454 3699 3307 3917 3292 3310 3283 3536
23JUL94 3756 3145 3571 3331 3725 3605 3547 3421 3257 3574
;
run;
```

A partial listing of TURBINE is shown in Figure 39.1. This data set is said to be in "strung-out" form since each observation contains the day and power output for a single heating. The first 20 observations contain the outputs for the first day, the second 20 observations contain the outputs for the second day, and so on. Because the variable DAY classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable KWATTS contains the output measurements and is referred to as the *process variable* (or *process* for short).

Kilowatt Power Output Data

Obs	day	kwatts
1	04JUL	3196
2	04JUL	3507
3	04JUL	3050
4	04JUL	3215
5	04JUL	3583
.	.	.
.	.	.
.	.	.
396	23JUL	3605
397	23JUL	3547
398	23JUL	3421
399	23JUL	3257
400	23JUL	3574

Figure 39.1. Partial Listing of the Data Set TURBINE

You can use a box chart to examine the distribution of power output for each day and to determine whether the mean level of the heating process is in control. The following statements create the box chart shown in Figure 39.2:

```
symbol h = .8;
title 'Box Chart for Power Output';
proc shewhart data=turbine;
   boxchart kwatts*day;
run;
```

This example illustrates the basic form of the BOXCHART statement. After the keyword BOXCHART, you specify the *process* to analyze (in this case, KWATTS), followed by an asterisk and the *subgroup-variable* (DAY).

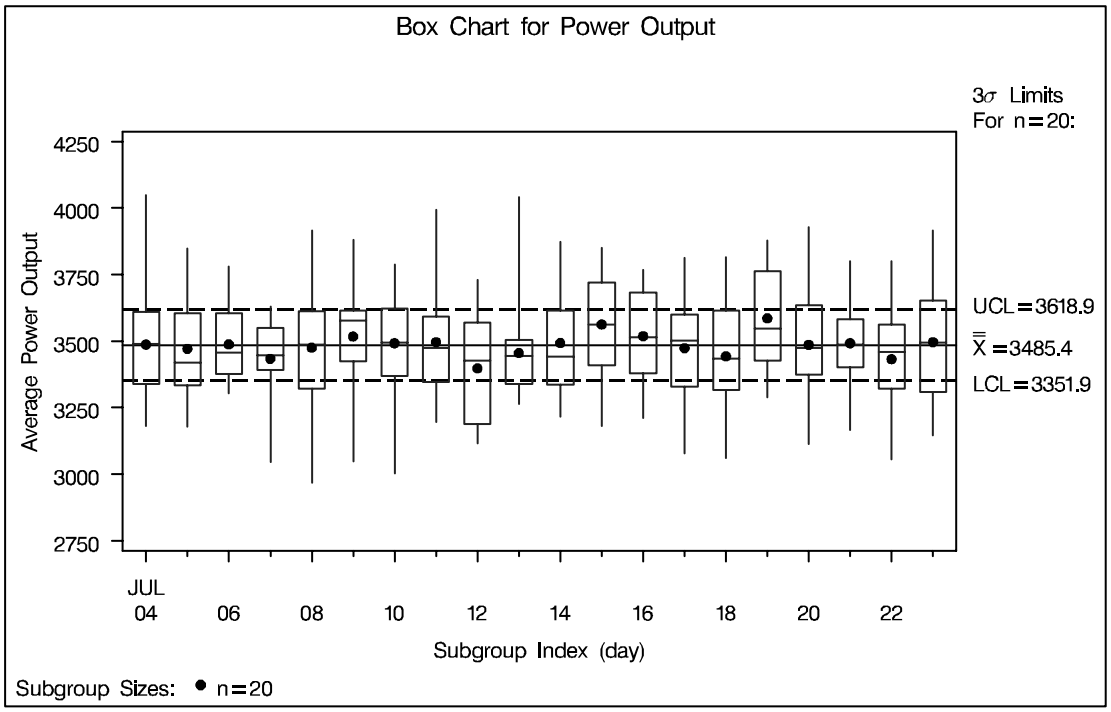


Figure 39.2. Box Chart for Power Output Data

The input data set is specified with the DATA= option in the PROC SHEWHART statement.

By default, the BOXCHART statement requests an $\bar{X}$ chart superimposed with box-and-whisker plots for each subgroup. Table 39.1 lists the summary statistics represented by each plot. For details on the computation of percentiles, see "Percentile Definitions" on page 1282.

Table 39.1. Summary Statistics Represented by Box-and-Whisker Plots

Subgroup Summary Statistic	Feature of Box-and-Whisker Plot
Maximum	Endpoint of upper whisker
Third quartile (75^{th} percentile)	Upper edge of box
Median (50^{th} percentile)	Line inside box
Mean	Symbol marker (in this example, a dot)
First quartile (25^{th} percentile)	Lower edge of box
Minimum	Endpoint of lower whisker

The within-subgroup variation in power output is stable, as indicated in Figure 39.2 by the edges of the boxes and the endpoints of the whiskers. Since the subgroup means, indicated by the dots, lie within the control limits, you can conclude that the heating process is in statistical control.

The skeletal style of the box-and-whisker plots shown in Figure 39.2 is the default. You can request different styles, as illustrated in Example 39.2 on page 1287. By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 39.24 on page 1267 and Table 39.25 on page 1268.

You can also create box charts in which the control limits apply to the subgroup medians. For example, the following statements create the chart shown in Figure 39.3:

```
symbol h = .8;
title 'Box Chart for Power Output';
proc shewhart data=turbine;
   boxchart kwatts*day / controlstat = median;
run;
```

The CONTROLSTAT=MEDIAN option requests control limits that apply to the medians. Alternatively, you can specify the NOLIMITS option to suppress the display of control limits and create ordinary side-by-side box-and-whisker plots. See Example 39.2 on page 1287.

Options such as CONTROLSTAT= and NOLIMITS are specified after the slash (/) in the BOXCHART statement. A complete list of options is presented in the "Syntax" section on page 1252.

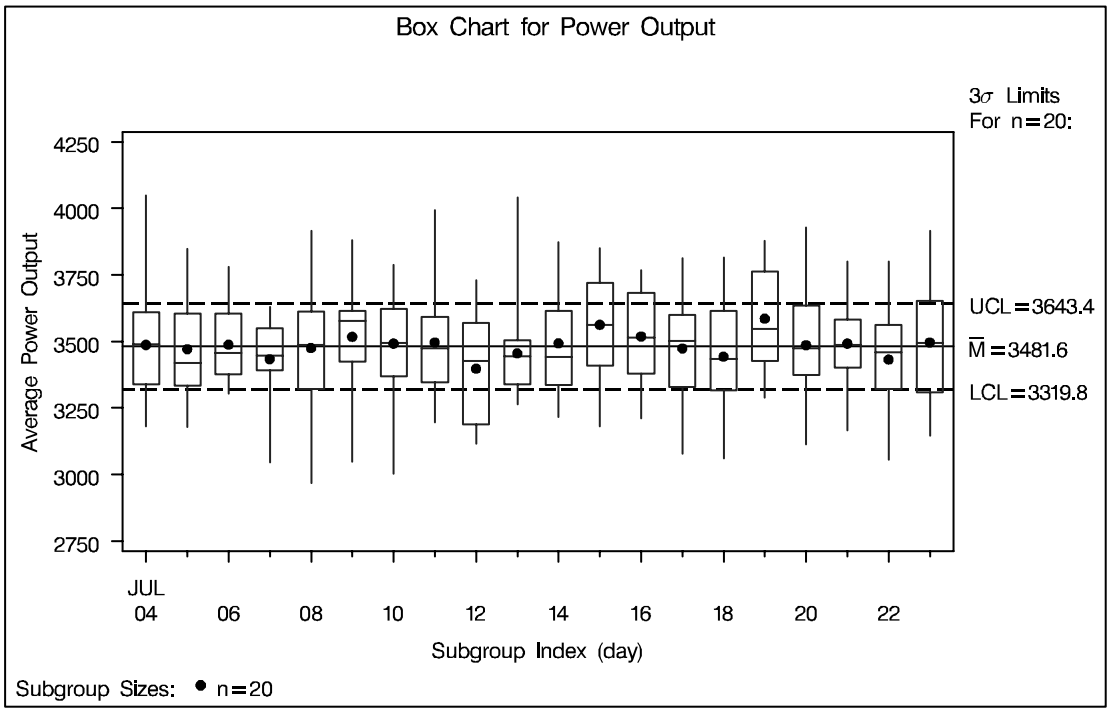


Figure 39.3. Box Chart for Power Output Data

Creating Box Charts from Subgroup Summary Data

The previous example illustrates how you can create box charts using raw data (process measurements). However, in many applications the data are provided as subgroup summary statistics. This example illustrates how you can use the BOXCHART statement with data of this type.

See SHWBOXA in the SAS/QC Sample Library

The following data set (OILSUM) provides the data from the preceding example in summarized form. There is exactly one observation for each subgroup (note that the subgroups are still indexed by DAY).

```
data oilsum;
   input day kwatts1 kwatts1 kwattsx kwattsm
             kwatts3 kwattsh kwattsr kwattsn;
   informat day date7. ;
   format day date5. ;
   label day    ='Date of Measurement'
         kwatts1='Minimum Power Output'
         kwatts1='25th Percentile'
         kwattsx='Average Power Output'
         kwattsm='Median Power Output'
         kwatts3='75th Percentile'
         kwattsh='Maximum Power Output'
         kwattsr='Range of Power Output'
         kwattsn='Subgroup Sample Size';
   datalines;
```

```
04JUL94 3180 3340.0 3487.40 3490.0 3610.0 4050 870 20
05JUL94 3179 3333.5 3471.65 3419.5 3605.0 3849 670 20
06JUL94 3304 3376.0 3488.30 3456.5 3604.5 3781 477 20
07JUL94 3045 3390.5 3434.20 3447.0 3550.0 3629 584 20
08JUL94 2968 3321.0 3475.80 3487.0 3611.5 3916 948 20
09JUL94 3047 3425.5 3518.10 3576.0 3615.0 3881 834 20
10JUL94 3002 3368.5 3492.65 3495.5 3621.5 3787 785 20
11JUL94 3196 3346.0 3496.40 3473.5 3592.5 3994 798 20
12JUL94 3115 3188.5 3398.50 3426.0 3568.5 3731 616 20
13JUL94 3263 3340.0 3456.05 3444.0 3505.5 4040 777 20
14JUL94 3215 3336.0 3493.60 3441.5 3616.0 3872 657 20
15JUL94 3182 3409.5 3563.30 3561.0 3719.5 3850 668 20
16JUL94 3212 3378.0 3519.05 3515.0 3682.5 3769 557 20
17JUL94 3077 3329.0 3474.20 3501.5 3599.5 3812 735 20
18JUL94 3061 3315.5 3443.60 3435.0 3614.5 3815 754 20
19JUL94 3288 3426.5 3586.35 3546.0 3762.5 3877 589 20
20JUL94 3114 3373.0 3486.45 3474.5 3635.5 3928 814 20
21JUL94 3167 3400.5 3492.90 3488.0 3582.5 3801 634 20
22JUL94 3056 3322.0 3432.80 3460.0 3561.0 3800 744 20
23JUL94 3145 3308.5 3496.90 3495.0 3652.0 3917 772 20
;
run;
```

A partial listing of OILSUM is shown in Figure 39.4.

Summary Data Set for Power Outputs

day	kwatts1	kwatts1	kwattsx	kwattsm	kwatts3	kwattsh	kwattsr	kwattsn
04JUL	3180	3340.0	3487.40	3490.0	3610.0	4050	870	20
05JUL	3179	3333.5	3471.65	3419.5	3605.0	3849	670	20
06JUL	3304	3376.0	3488.30	3456.5	3604.5	3781	477	20
07JUL	3045	3390.5	3434.20	3447.0	3550.0	3629	584	20
08JUL	2968	3321.0	3475.80	3487.0	3611.5	3916	948	20
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.

Figure 39.4. The Summary Data Set OILSUM

There are eight summary variables in OILSUM.

- KWATTSL contains the subgroup minimums (low values).
- KWATTS1 contains the 25^{th} percentile (first quartile) for each subgroup.
- KWATTSX contains the subgroup means.
- KWATTSM contains the subgroup medians.
- KWATTS3 contains the 75^{th} percentile (third quartile) for each subgroup.
- KWATTSH contains the subgroup maximums (high values).
- KWATTSR contains the subgroup ranges.
- KWATTSN contains the subgroup sample sizes.

You can read this data set by specifying it as a HISTORY= data set in the PROC

SHEWHART statement, as illustrated by the following statements, which create the box chart shown in Figure 39.5:

```
symbol h = .8;
title 'Box Chart for Power Output';
proc shewhart history=oilsum;
   boxchart kwatts*day / ranges;
run;
```

Note that the *process* KWATTS is *not* the name of a SAS variable in the data set but is, instead, the common prefix for the names of the eight summary variables. The suffix characters L, 1, X, M, 3, H, R, and N indicate the contents of the variable. For example, the suffix characters 1 and 3 indicate first and third quartiles. The name DAY specified after the asterisk is the name of the *subgroup-variable*.

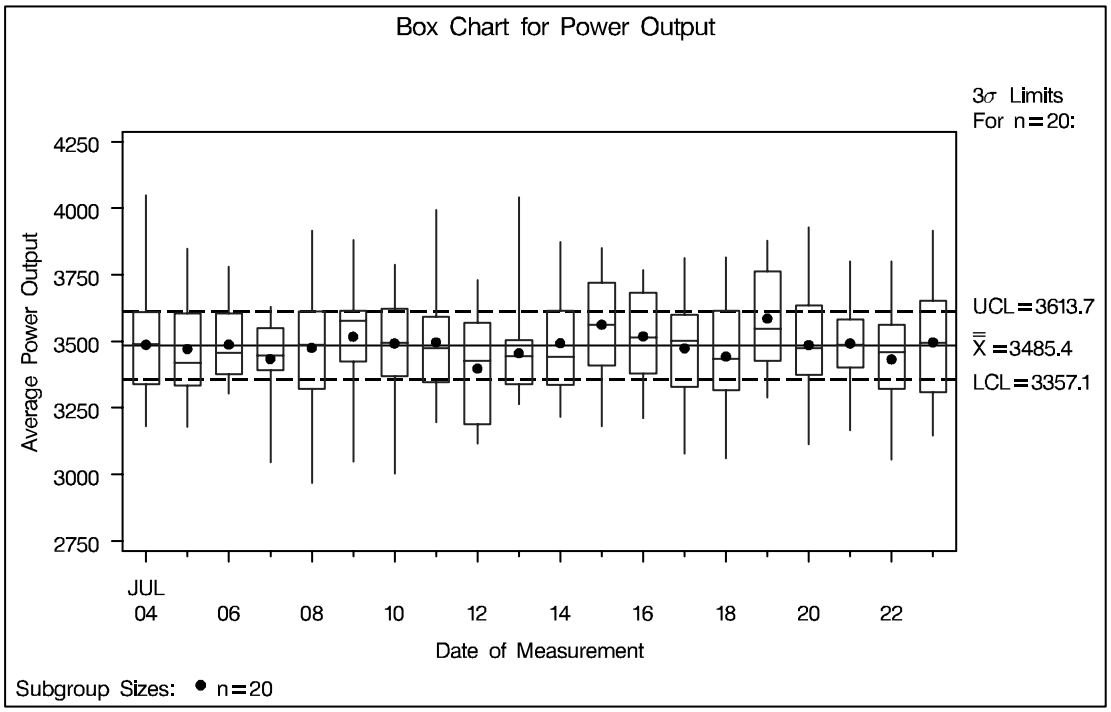


Figure 39.5. Box Chart for Power Output Data

In general, a HISTORY= input data set used with the BOXCHART statement must contain the following variables:

- subgroup variable
- subgroup minimum variable
- subgroup first quartile variable
- subgroup mean variable
- subgroup median variable
- subgroup third quartile variable
- subgroup maximum variable

- subgroup sample size variable
- either a subgroup standard deviation variable or a subgroup range variable

Furthermore, the names of the summary variables must begin with the *process* name specified in the BOXCHART statement and end with the appropriate suffix character. If the names do not follow this convention, you can use the RENAME option in the PROC SHEWHART statement to rename the variables for the duration of the SHEWHART procedure step (see page 1743).

If you specify the RANGES option in the BOXCHART statement, the HISTORY= data set must contain a subgroup range variable; otherwise, the HISTORY= data set must contain a subgroup standard deviation variable. The RANGES option specifies that the estimate of the process standard deviation σ is to be calculated from subgroup ranges rather than subgroup standard deviations. For example, in the following statements, the data set OILSUM2 must contain a subgroup standard deviation variable named KWATTSS, because the RANGES option not specified:

```
title 'Box Chart for Power Output';
symbol v=dot;
proc shewhart history=oilsum2;
   boxchart kwatts*day;
run;
```

In summary, the interpretation of *process* depends on the input data set.

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names of the variables containing the summary statistics.

For more information, see "HISTORY= Data Set" on page 1276.

Saving Summary Statistics

See SHWBOXA in the SAS/QC Sample Library

In this example, the BOXCHART statement is used to create a summary data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read measurements from the data set TURBINE and create a summary data set named TURBHIST:

```
title 'Summary Data Set for Power Output';
proc shewhart data=turbine;
   boxchart kwatts*day / outhistory = turbhist
                         nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 39.2.

Figure 39.6 contains a partial listing of TURBHIST.

Summary Data Set for Power Output

day	kwatts L	kwatts1	kwattsX	kwatts M	kwatts3	kwatts H	kwattsS	kwatts N
04JUL	3180	3340.0	3487.40	3490.0	3610.0	4050	220.260	20
05JUL	3179	3333.5	3471.65	3419.5	3605.0	3849	210.427	20
06JUL	3304	3376.0	3488.30	3456.5	3604.5	3781	147.025	20
07JUL	3045	3390.5	3434.20	3447.0	3550.0	3629	157.637	20
08JUL	2968	3321.0	3475.80	3487.0	3611.5	3916	258.949	20
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.

Figure 39.6. The Summary Data Set TURBHIST

There are nine variables in the data set TURBHIST.

- DAY is the subgroup variable.
- KWATTSL contains the subgroup minimums.
- KWATTS1 contains the first quartiles for each subgroup.
- KWATTSX contains the subgroup means.
- KWATTSM contains the subgroup medians.
- KWATTS3 contains the third quartiles for each subgroup.
- KWATTSH contains the subgroup maximums.
- KWATTSS contains the subgroup standard deviations.
- KWATTSN contains the subgroup sample sizes.

Note that the summary statistic variables are named by adding the suffix characters *L*, *1*, *X*, *M*, *3*, *H*, *S*, and *N* to the *process* KWATTS specified in the BOXCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

If you specify the RANGES option, the OUTHISTORY= data set includes a subgroup range variable, rather than a subgroup standard deviation variable, as demonstrated by the following statements:

```
proc shewhart data=turbine;
   boxchart kwatts*day / outhistory = turbhist2
                         ranges
                         nochart;
run;
```

Figure 39.7 contains a partial listing of TURBHIST2. The variable KWATTSR contains the subgroup ranges.

The RANGES option is not recommended when the subgroup sample sizes are greater than 10, nor when you use the NOLIMITS option to create standard side-by-side box-and-whisker plots.

For more information, see "OUTHISTORY= Data Set" on page 1271.

Summary Data Set for Power Output

day	kwatts L	kwatts1	kwattsX	kwatts M	kwatts3	kwatts H	kwatts R	kwatts N
04JUL	3180	3340.0	3487.40	3490.0	3610.0	4050	870	20
05JUL	3179	3333.5	3471.65	3419.5	3605.0	3849	670	20
06JUL	3304	3376.0	3488.30	3456.5	3604.5	3781	477	20
07JUL	3045	3390.5	3434.20	3447.0	3550.0	3629	584	20
08JUL	2968	3321.0	3475.80	3487.0	3611.5	3916	948	20
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.

Figure 39.7. The Summary Data Set TURBHIST2

Saving Control Limits

See SHWBOXA in the SAS/QC Sample Library

You can save the control limits for a box chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1251) or modify the limits with a DATA step program.

The following statements read measurements from the data set TURBINE (see page 1240) and save the control limits displayed in Figure 39.2 in a data set named TURBLIM:

```
proc shewhart data=turbine;
   boxchart kwatts*day / outlimits=turblim
                         nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. The data set TURBLIM is listed in Figure 39.8.

Control Limits for Power Output Data

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_
kwatts	day	ESTIMATE	20	.002699796	3	3351.92

MEAN	_UCLX_	_LCLS_	_S_	_UCLS_	_STDDEV_
3485.41	3618.90	100.207	196.396	292.584	198.996

Figure 39.8. The Data Set TURBLIM Containing Control Limit Information

The data set TURBLIM contains one observation with the limits for *process* KWATTS. The variables _LCLX_ and _UCLX_ contain the lower and upper control limits for the means, and the variable _MEAN_ contains the central line. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values.

The variables _LCLS_, _S_, and _UCLS_ are not used to create box charts, but they are included so that the data set TURBLIM can be used to create an s chart; see Chapter 51, "XSCHART Statement," . If you specify the RANGES option in the BOXCHART statement, the variables _LCLR_, _R_, and _UCLR_, rather than the variables _LCLS_, _S_, and _UCLS_, are included in the OUTLIMITS= data set. These variables can be used to create an R chart; see Chapter 50, "XRCHART Statement."

If you specify CONTROLSTAT=MEDIAN to request control limits for medians, the variables _LCLM_ and _UCLM_, rather than the variables _LCLX_ and _UCLX_, are included in the OUTLIMITS= data set as demonstrated by the following statements:

```
proc shewhart data=turbine;
   boxchart kwatts*day / outlimits   = turblim2
                         controlstat = median
                         nochart;
run;
```

TURBLIM2 is listed in Figure 39.9. For more information, see "OUTLIMITS= Data Set" on page 1269.

Control Limits for Power Output Data

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLM_
kwatts	day	ESTIMATE	20	.002776264	3	3319.85

MEAN	_UCLM_	_LCLS_	_S_	_UCLS_	_STDDEV_
3481.63	3643.40	100.207	196.396	292.584	198.996

Figure 39.9. The Data Set TURBLIM2 Containing Control Limit Information

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
title 'Summary Statistics and Control Limit Information';
proc shewhart data=turbine;
   boxchart kwatts*day / outtable=turbtab
                         nochart;
run;
```

The data set TURBTAB is partially listed in Figure 39.10.

Summary Statistics and Control Limit Information

VAR	day	_SIGMAS_	_LIMITN_	_SUBN_	_LCLX_	_SUBX_	_MEAN_
kwatts	04JUL	3	20	20	3351.92	3487.40	3485.41
kwatts	05JUL	3	20	20	3351.92	3471.65	3485.41
kwatts	06JUL	3	20	20	3351.92	3488.30	3485.41
kwatts	07JUL	3	20	20	3351.92	3434.20	3485.41
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.

UCLX	_EXLIM_	_SUBMIN_	_SUBQ1_	_SUBMED_	_SUBQ3_	_SUBMAX_
3618.90		3180	3340.0	3490.0	3610.0	4050
3618.90		3179	3333.5	3419.5	3605.0	3849
3618.90		3304	3376.0	3456.5	3604.5	3781
3618.90		3045	3390.5	3447.0	3550.0	3629
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.

Figure 39.10. The OUTTABLE= Data Set TURBTAB

This data set contains one observation for each subgroup sample. The variable _SUBMIN_ contains the subgroup minimums, and the variable _SUBQ1_ contains the first quartile for each subgroup. The variable _SUBX_ contains the subgroup means, and the variable _SUBMED_ contains the subgroup medians. The variable _SUBQ3_ contains the third quartiles, and the variable _SUBMAX_ contains the subgroup maximums. The variable _SUBN_ contains the subgroup sample sizes. The variables _LCLX_ and _UCLX_ contain the lower and upper control limits for the means. The variable _MEAN_ contains the central line. The variables _VAR_ and DAY contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1273.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read TURBTAB and display a box chart (not shown here) identical to the chart in Figure 39.2:

```
title 'Box Chart for Power Output';
symbol v=dot;
proc shewhart table=turbtab;
   boxchart kwatts*day;
label _SUBX_ = 'Average Power Output';
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see Chapter 56, "Specialized Control Charts,").

For more information, see "TABLE= Data Set" on page 1277.

Reading Preestablished Control Limits

See SHWBOXA in the SAS/QC Sample Library

In the previous example, the OUTLIMITS= data set TURBLIM saved control limits computed from the measurements in TURBINE. This example shows how these limits can be applied to new data. The following statements create the box chart in Figure 39.11 using new measurements in a data set named TURBINE2 (not listed here) and the control limits in TURBLIM:

```
symbol h = .8;
title 'Box Chart for Power Output';
proc shewhart data=turbine2 limits=turblim;
   boxchart kwatts*day;
run;
```

The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default,* this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name KWATTS
- the value of _SUBGRP_ matches the *subgroup-variable* name DAY

The chart reveals an increase in variability beginning on August 1.

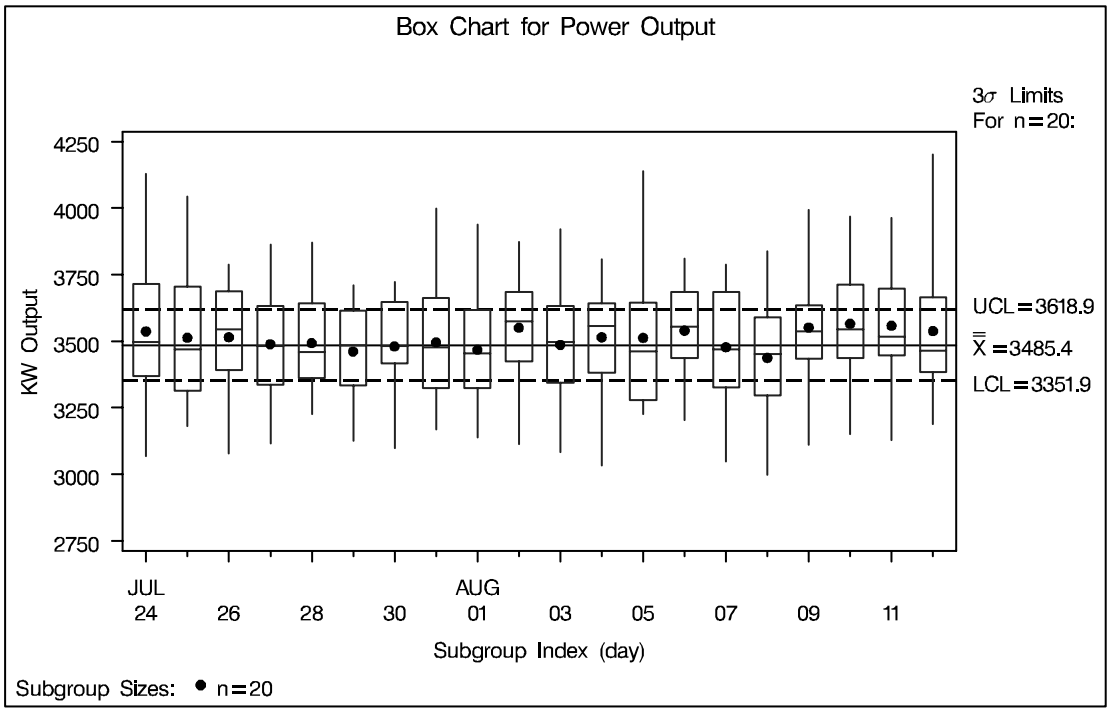


Figure 39.11. Box Chart for Second Set of Power Outputs

*In Release 6.09 and in earlier releases, it is also necessary to specify the READLIMITS option to read control limits from a LIMITS= data set.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1275 for details concerning the variables that you must provide.

Syntax

The basic syntax for the BOXCHART statement is as follows:

BOXCHART *process*******subgroup-variable* **;**

The general form of this syntax is as follows:

BOXCHART (*processes***)****subgroup-variable* <**(***block-variables* **)** >
< =*symbol-variable* | =*'character'* > < **/** *options* >**;**

You can use any number of BOXCHART statements in the SHEWHART procedure. The components of the BOXCHART statement are described as follows.

process
processes
identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating Box Charts from Raw Data" on page 1240.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating Box Charts from Subgroup Summary Data" on page 1243.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1248.

A *process* is required. If you specify more than one *process*, enclose the list in parentheses. For example, the following statements request distinct box charts for WEIGHT, LENGTH, and WIDTH:

```
proc shewhart data=summary;
   boxchart (weight length width)*day;
run;
```

subgroup-variable
is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding BOXCHART statement, DAY is the subgroup variable. For details, see "Subgroup Variables" on page 1771.

block-variables
are optional variables that group the data into blocks of consecutive subgroups. These blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 1932 for an example.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the means.

- If you produce a chart on a line printer, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce a chart on a graphics device, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOLn statements. See "Displaying Stratification in Levels of a Classification Variable" on page 1931 for an example.

character
specifies a plotting character for charts produced on line printers. For example, the following statements create a box chart using an asterisk (*) to plot the means:

```
proc shewhart data=values;
   boxchart weight*day='*';
run;
```

options
enhance the appearance of the box chart, request additional analyses, save results in data sets, and so on. The "Summary of Options" section, which follows, lists all options by function. Chapter 53, "Dictionary of Options," describes each option in detail.

Summary of Options

The following tables list the BOXCHART statement options by function. For complete descriptions, see Chapter 53, "Dictionary of Options."

Table 39.2. Tabulation Options

TABLE	creates a basic table of subgroup values, subgroup sample sizes, subgroup summary statistics, and control limits
TABLEALL	is equivalent to the options TABLE, TABLEBOX, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUT, and TABLETEST
TABLEBOX	augments basic table with columns for minimum, 25^{th} percentile, median, 75^{th} percentile, and maximum of observations in a subgroup
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with a column indicating which tests for special causes are positive

Note that specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.

Table 39.3. Options for Controlling Box Appearance

BOXCONNECT	connects subgroup means in box-and-whisker plots
BOXCONNECT=*keyword*	connects subgroup means, medians, maximum values, minimum values, or quartiles in box-and-whisker plots
BOXSTYLE=*keyword*	specifies style of box-and-whisker plots
BOXWIDTH=*value*	specifies width of box-and-whisker plots
BOXWIDTHSCALE=*value*	specifies that widths of box-and-whisker plots vary proportionately to subgroup sample size
CBOXES=*color*|(*variable*)	specifies color for outlines of box-and-whisker plots
CBOXFILL=*color*|(*variable*)	specifies fill color for interior of box-and-whisker plots
IDCOLOR=*color*	specifies outlier symbol color in schematic box-and-whisker plots
IDCTEXT=*color*	specifies text color to label outliers or process variable values
IDFONT=*font*	specifies text font to label outliers or process variable values
IDHEIGHT=*value*	specifies text height to label outliers or process variable values
IDSYMBOL=*symbol*	specifies outlier symbol in schematic box-and-whisker plots
LBOXES=*linetype*|(*variable*)	specifies line types for outlines of box-and-whisker plots
NOTCHES	specifies that box-and-whisker plots are to be notched
PCTLDEF=*n*	specifies percentile definition used for box-and-whisker plots
SERIFS	adds serifs to the whiskers of skeletal box-and-whisker plots

Table 39.4. Options for Specifying Tests for Special Causes

NO3SIGMACHECK	enables tests to be applied with control limits other than 3σ limits
TESTS=*value-list*| *customized-pattern-list*	specifies tests for special causes for the box chart
TEST2RUN=*n*	specifies length of pattern for Test 2
TEST3RUN=*n*	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL='*label*'| (*variable*)|*keyword*	provides labels for points where test is positive
TESTLABEL*n*='*label*'	specifies label for n^{th} test for special causes
TESTNMETHOD= STANDARDIZE	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
TESTRESET=*variable*	enables tests for special causes to be reset for the box chart
ZONES	adds lines to box chart delineating zones A, B, and C
ZONEVALPOS=*n*	specifies position of ZONEVALUES labels
ZONEVALUES	labels zone lines with their values

Table 39.5. Line Printer Options for Displaying Tests for Special Causes

TESTCHAR='*character*'	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
ZONECHAR='*character*'	specifies character for lines that delineate zones for tests for special causes

Table 39.6. Graphical Options for Displaying Tests for Special Causes

CTESTLABBOX=*color*	specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=*color*| *test-color-list*	specifies color for labels indicating points where test is positive
CTESTSYMBOL=*color*	specifies color for symbol used to plot points where test is positive
CZONES=*color*	specifies color for lines and labels delineating zones A, B, and C
LTESTS=*linetype*	specifies type of line connecting points where test is positive
LZONES=*linetype*	specifies line type for lines delineating zones A, B, and C
TESTFONT=*font*	specifies software font for labels at points where test is positive
TESTHEIGHT=*value*	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=*symbol*	specifies plot symbol for points where test is positive
TESTSYMBOLHT=*value*	specifies symbol height for points where test is positive
WTESTS=*n*	specifies width of line connecting points where test is positive

Table 39.7. Options for Interactive Control Charts

HTML=(*variable*)	specifies variable whose values are URLs to be associated with subgroups on primary chart
HTML2=(*variable*)	specifies variable whose values are URLs to be associated with subgroups on secondary chart
HTML_LEGEND=(*variable*)	specifies variable whose values are URLs to be associated with symbols in the symbol legend
OUTHIGHHTML=(*variable*)	specifies variable whose values are URLs to be associated with outliers above the upper fence on a schematic box chart
OUTLOWHTML=(*variable*)	specifies variable whose values are URLs to be associated with outliers below the lower fence on a schematic box chart
POINTSHTML=(*variable*)	specifies variable whose values are URLs to be associated with points representing individual observations
WEBOUT=*SAS-data-set*	creates an OUTTABLE= data set with additional graphics coordinate data

Table 39.8. Clipping Options

CCLIP=*color*	specifies color for plot symbol for clipped points
CLIPCHAR='*character*'	specifies plot character for clipped points
CLIPFACTOR=*value*	determines extent to which extreme points are clipped
CLIPLEGEND='*string*'	specifies text for clipping legend
CLIPLEGPOS=*keyword*	specifies position of clipping legend
CLIPSUBCHAR='*character*'	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=*symbol*	specifies plot symbol for clipped points
CLIPSYMBOLHT=*value*	specifies symbol marker height for clipped points

Table 39.9. Options for Plotting and Labeling Points

ALLLABEL=VALUE|(*variable*)	labels every point on box chart
ALLLABEL2=VALUE|(*variable*)	labels every point on trend chart
CCONNECT=*color*	specifies color for line segments that connect points on chart
CFRAMELAB=*color*	specifies fill color for frame around labeled points
CLABEL=*color*	specifies color for labels
CONNECTCHAR='*character*'	specifies character used to form line segments that connect points on chart
COUT=*color*	specifies color for portions of line segments that connect points outside control limits
COUTFILL=*color*	specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=*angle*	specifies angle at which labels are drawn
LABELFONT=*font*	specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=*value*	specifies height of labels (alias for the TESTHEIGHT= option)
NOCONNECT	suppresses line segments that connect points on chart
NOTRENDCONNECT	suppresses line segments that connect points on trend chart
OUTLABEL=VALUE|(*variable*)	labels points outside control limits on box chart
SYMBOLCHARS='*characters*'	specifies characters indicating *symbol-variable*
SYMBOLLEGEND=NONE|*name*	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=*keyword*	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically

Table 39.10. Block Variable Legend Options

BLOCKLABELPOS=*keyword*	specifies position of label for *block-variable* legend
BLOCKLABTYPE=*n*|*keyword*	specifies text size of *block-variable* legend
BLOCKPOS=*n*	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=*color-list*	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=*variable*|(*variables*)	specifies one or more variables whose values are colors for filling background of *block-variable* legend

Table 39.11. Reference Line Options

CHREF=*color*	specifies color for lines requested by HREF= and HREF2= options
CVREF=*color*	specifies color for lines requested by VREF= and VREF2= options
HREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on box chart
HREF2=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFCHAR='*character*'	specifies line character for HREF= and HREF2= lines
HREFDATA= *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on box chart
HREF2DATA= *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFLABELS= (*'label1'...'labeln'*)	specifies labels for HREF= lines
HREF2LABELS= (*'label1'...'labeln'*)	specifies labels for HREF2= lines
HREFLABPOS=*n*	specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=*linetype*	specifies line type for HREF= and HREF2= lines
LVREF=*linetype*	specifies line type for VREF= and VREF2= lines
NOBYREF	specifies that reference line information in a data set is to be applied uniformly to charts created for all BY groups
VREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to vertical axis on box chart
VREF2=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to vertical axis on trend chart
VREFCHAR='*character*'	specifies line character for VREF= and VREF2= lines
VREFLABELS= *'label1'...'labeln'*	specifies labels for VREF= lines
VREF2LABELS= *'label1'...'labeln'*	specifies labels for VREF2= lines
VREFLABPOS=*n*	specifies position of VREFLABELS= and VREF2LABELS= labels

Table 39.12. Grid Options

CGRID=*color*	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=*linetype*	specifies line type for grid requested with the ENDGRID option
LGRID=*linetype*	specifies line type for grid requested with the GRID option
WGRID=*n*	specifies width of grid lines

Table 39.13. Options for Specifying Control Limits

ALPHA=*value*	requests probability limits for control charts
CONTROLSTAT=*keyword*	specifies whether control limits are computed for subgroup means or subgroup medians
LIMITN=*n*|VARYING	specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	computes control limits for each *process* from the data rather than from a LIMITS= data set (Release 6.10 and later releases)
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEXES=ALL|'*label1*'...'*labeln*'	reads multiple sets of control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (Release 6.09 and earlier releases)
SIGMAS=*k*	specifies width of control limits in terms of multiple *k* of standard error of plotted statistic

Table 39.14. Options for Displaying Control Limits

CINFILL=*color*	specifies color for area inside control limits
CLIMITS=*color*	specifies color of control limits, central line, and related labels
LCLLABEL='*label*'	specifies label for lower control limit on box chart
LIMLABSUBCHAR='*character*'	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=*linetype*	specifies line type for control limits
NDECIMAL=*n*	specifies number of digits to right of decimal place in default labels for control limits and central line in box chart
NOCTL	suppresses display of central line in box chart
NOLCL	suppresses display of lower control limit in box chart
NOLIMITLABEL	suppresses labels for control limits and central line
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOUCL	suppresses display of upper control limit in box chart
UCLLABEL='*string*'	specifies label for upper control limit in box chart
WLIMITS=*n*	specifies width for control limits and central line
XSYMBOL='*string*'|*keyword*	specifies label for central line in box chart

Table 39.15. Axis and Axis Label Options

CAXIS=*color*	specifies color for axis lines and tick marks
CFRAME=*color*|(*color-list*)	specifies fill colors for frame for plot area
CTEXT=*color*	specifies color for tick mark values and axis labels
HAXIS=*values*|AXIS*n*	specifies major tick mark values for horizontal axis
HEIGHT=*value*	specifies height of axis label and axis legend text
HMINOR=*n*	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=*value*	specifies length of offset at both ends of horizontal axis
INTSTART=*value*	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=*n*	specifies thinning factor for tick mark labels on horizontal axis
SPLIT='*character*'	specifies splitting character for axis labels
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=*values*|AXIS*n*	specifies major tick mark values for vertical axis of box chart
VAXIS2=*values*|AXIS*n*	specifies major tick mark values for vertical axis of trend chart
VFORMAT=*format*	specifies format for primary vertical axis tick mark labels
VFORMAT2=*format*	specifies format for secondary vertical axis tick mark labels
VMINOR=*n*	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=*value*	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis for primary chart
VZERO2	forces origin to be included in vertical axis for secondary chart
WAXIS=*n*	specifies width of axis lines

Table 39.16. Input Data Set Options

MISSBREAK	specifies that observations with missing values are not to be processed

Table 39.17. Output Data Set Options

OUTBOX= *SAS-data-set*	creates output data set containing subgroup summary statistics, control limits, and outlier values for box chart
OUTHISTORY= *SAS-data-set*	creates output data set containing subgroup summary statistics
OUTINDEX='*string*'	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS= *SAS-data-set*	creates output data set containing control limits
OUTTABLE= *SAS-data-set*	creates output data set containing subgroup summary statistics and control limits

Table 39.18. Phase Options

CPHASEBOX=*color*	specifies color for box enclosing all plotted points for a phase
CPHASEBOX-CONNECT=*color*	specifies color for line segments connecting adjacent enclosing boxes
CPHASEBOXFILL=*color*	specifies fill color for box enclosing all plotted points for a phase
CPHASELEG=*color*	specifies text color for *phase* legend
CPHASEMEAN-CONNECT=*color*	specifies color for line segments connecting average value points within a phase
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE='*string*'	specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=*value*| *keyword*	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEMEANSYMBOL= *symbol*	specifies symbol marker for average of values within a phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES= ALL| '*label1*'...'*labeln*'	specifies *phases* to be read from an input data set

Table 39.19. Specification Limit Options

CIINDICES (ALPHA=*value* TYPE=*keyword*)	specifies α value and type for computing capability index confidence limits
LSL=*value-list*	specifies list of lower specification limits
TARGET=*value-list*	specifies list of target values
USL=*value-list*	specifies list of upper specification limits

Table 39.20. Process Mean and Standard Deviation Options

MEDCENTRAL=*keyword*	specifies method for estimating process mean μ
MU0=*value*	specifies known value of μ_0 for process mean μ
RANGES	specifies that estimate of process standard deviation σ is to be calculated from subgroup ranges
SIGMA0=*value*	specifies known value σ_0 for process standard deviation σ
SMETHOD=*keyword*	specifies method for estimating process standard deviation σ
TYPE=*keyword*	identifies whether parameters are estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set

Table 39.21. Graphical Enhancement Options

ANNOTATE= *SAS-data-set*	specifies annotate data set that adds features to box chart
ANNOTATE2= *SAS-data-set*	specifies annotate data set that adds features to trend chart
DESCRIPTION='*string*'	specifies string that appears in the description field of the PROC GREPLAY master menu for box chart
FONT=*font*	specifies software font for labels and legends on charts
NAME='*string*'	specifies name that appears in the name field of the PROC GREPLAY master menu for box chart
PAGENUM='*string*'	specifies the form of the label used in pagination
PAGENUMPOS= *keyword*	specifies the position of the page number requested with the PAGENUM= option
WTREND=*n*	specifies width of line segments connecting points on trend chart

Table 39.22. Plot Layout Options

ALLN	plots summary statistics for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts for a process variable only when exceptions occur
INTERVAL=*keyword*	specifies natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=*n*	specifies maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of box chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=*n*	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=*n*	specifies number of pages or screens to be used to display chart
TRENDVAR=*variable*|(*variable-list*)	specifies list of trend variables
YPCT1=*value*	specifies length of vertical axis on box chart as a percentage of sum of lengths of vertical axes for box and trend charts
ZEROSTD	displays box chart regardless of whether $\hat{\sigma} = 0$

Table 39.23. Overlay Options

CCOVERLAY=*color-list*	specifies colors for primary chart overlay line segments
CCOVERLAY2=*color-list*	specifies colors for secondary chart overlay line segments
COVERLAY=*color-list*	specifies colors for primary chart overlay plots
COVERLAY2=*color-list*	specifies colors for secondary chart overlay plots
COVERLAYCLIP=*color*	specifies color for clipped points on overlays
LOVERLAY=*linetypes*	specifies line types for primary chart overlay line segments
LOVERLAY2=*linetypes*	specifies line types for secondary chart overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=*variable-list*	specifies variables to overlay on primary chart
OVERLAY2=*variable-list*	specifies variables to overlay on secondary chart
OVERLAY2HTML=*variable-list*	specifies URLS to associate with secondary chart overlay points
OVERLAY2ID=*variable-list*	specifies labels for secondary chart overlay points
OVERLAY2SYM=*symbol-list*	specifies symbols for secondary chart overlays
OVERLAY2SYMHT=*value-list*	specifies symbol heights for secondary chart overlays
OVERLAYCLIPSYM=*symbol*	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=*value*	specifies symbol height for clipped points on overlays
OVERLAYHTML=*variable-list*	specifies URLS to associate with primary chart overlay points
OVERLAYID=*variable-list*	specifies labels for primary chart overlay points
OVERLAYLEGLAB='*label*'	specifies label for overlay legend
OVERLAYSYM=*symbol-list*	specifies symbols for primary chart overlays
OVERLAYSYMHT=*value-list*	specifies symbol heights for primary chart overlays
WOVERLAY=*value-list*	specifies widths of primary chart overlay line segments
WOVERLAY2=*value-list*	specifies widths of secondary chart overlay line segments

Details

Constructing Box Charts

The following notation is used in this section:

μ	process mean (expected value of the population of measurements)
σ	process standard deviation (standard deviation of the population of measurements)
$\bar{X}_i$	mean of measurements in i^{th} subgroup
n_i	sample size of i^{th} subgroup
N	the number of subgroups
x_{ij}	j^{th} measurement in the i^{th} subgroup, $j = 1, 2, 3 \ldots, n_i$
$x_{i(j)}$	j^{th} largest measurement in the i^{th} subgroup: $$x_{i(1)} \leq x_{i(2)} \leq \ldots \leq x_{i(n_i)}$$
$\overline{\overline{X}}$	weighted average of subgroup means
M_i	median of the measurements in the i^{th} subgroup: $$M_i = \begin{cases} x_{i((n_i+1)/2)} & \text{if } n_i \text{ is odd} \\ (x_{i(n_i/2)} + x_{i((n_i/2)+1)})/2 & \text{if } n_i \text{ is even} \end{cases}$$
$\bar{M}$	average of the subgroup medians: $$\bar{M} = (n_1 M_1 + \ldots + n_N M_N)/(n_1 + \ldots + n_N)$$
$\tilde{M}$	median of the subgroup medians. Denote the j^{th} largest median by $M_{(j)}$ so that $M_{(1)} \leq M_{(2)} \leq \ldots \leq M_{(N)}$. $$\tilde{M} = \begin{cases} M_{((N+1)/2)} & \text{if } N \text{ is odd} \\ (M_{(N/2)} + M_{(N/2)+1})/2 & \text{if } N \text{ is even} \end{cases}$$
$e_M(n)$	standard error of the median of n independent, normally distributed variables with unit standard deviation (the value of $e_M(n)$ can be calculated with the STDMED function in a DATA step)
$Q_p(n)$	$100p^{\text{th}}$ percentile $(0 < p < 1)$ of the distribution of the median of n independent observations from a normal population with unit standard deviation
z_p	$100p^{\text{th}}$ percentile of the standard normal distribution
$D_p(n)$	$100p^{\text{th}}$ percentile of the distribution of the range of n independent observations from a normal population with unit standard deviation

Elements of Box-and-Whisker Plots

A box-and-whisker plot is displayed for the measurements in each subgroup on the box chart. Figure 39.12 illustrates the elements of each plot.

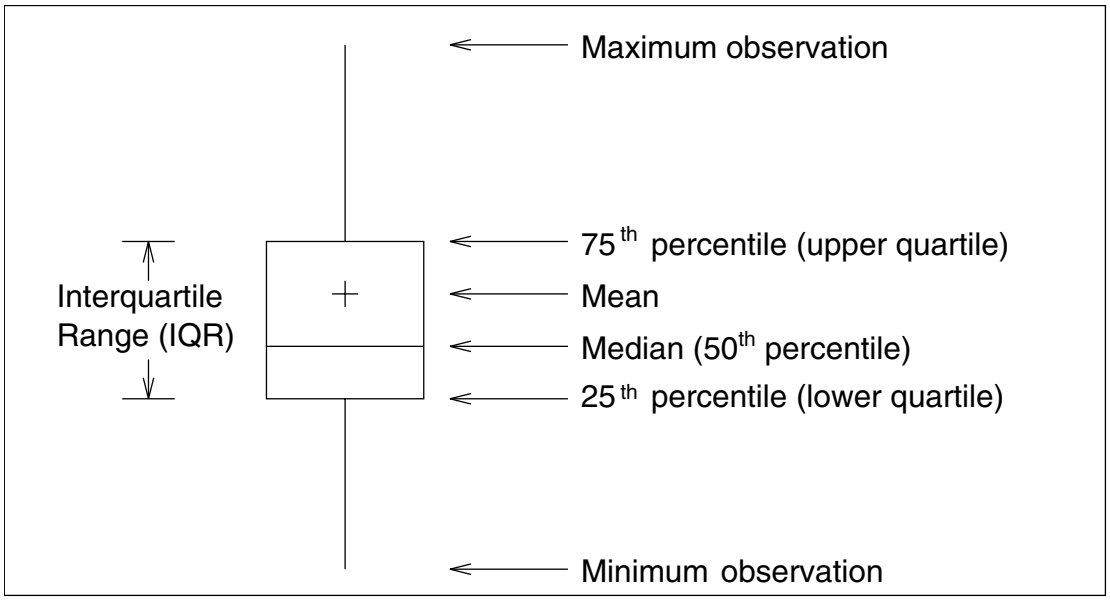


Figure 39.12. Box-and-Whisker Plot

The skeletal style of the box-and-whisker plot shown in Figure 39.12 is the default. You can specify alternative styles with the BOXSTYLE= option; see Example 39.2 on page 1287 or the entry for BOXSTYLE= on page 1858 in Chapter 53, "Dictionary of Options."

Control Limits and Central Line

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of $\bar{X}_i$ (or M_i) above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that $\bar{X}_i$ (or M_i) exceeds the limits

The CONTROLSTAT= option specifies whether control limits are computed for subgroup means (the default) or subgroup medians. The following tables provide the formulas for the limits:

Table 39.24. Control Limits and Central Line for Box Charts

CONTROLSTAT=MEAN	CONTROLSTAT=MEDIAN
LCLX = lower limit = $\overline{\overline{X}} - k\hat{\sigma}/\sqrt{n_i}$ Central Line = $\overline{\overline{X}}$ UCLX = upper limit = $\overline{\overline{X}} + k\hat{\sigma}/\sqrt{n_i}$	LCLM = lower limit = $\bar{M} - k\hat{\sigma}e_M(n_i)$ Central Line = $\bar{M}$ UCLM = upper limit = $\bar{M} + k\hat{\sigma}e_M(n_i)$

Table 39.25. Probability Limits and Central Line for Box Charts

CONTROLSTAT=MEAN	CONTROLSTAT=MEDIAN
LCLX = lower limit = $\overline{\overline{X}} - z_{\alpha/2}(\hat{\sigma}/\sqrt{n_i})$ Central Line = $\overline{\overline{X}}$ UCLX = upper limit = $\overline{\overline{X}} + z_{\alpha/2}(\hat{\sigma}/\sqrt{n_i})$	LCLM = lower limit = $\bar{M} - Q_{\alpha/2}(n_i)\hat{\sigma}$ Central Line = $\bar{M}$ UCLM = upper limit = $\bar{M} + Q_{1-\alpha/2}(n_i)\hat{\sigma}$

In the preceding tables, replace $\bar{M}$ with $\overline{\overline{X}}$ if you specify MEDCENTRAL=AVGMEAN in addition to CONTROLSTAT=MEDIAN. Likewise, replace $\bar{M}$ with $\tilde{M}$ if you specify MEDCENTRAL=MEDMED in addition to CONTROLSTAT=MEDIAN. If standard values μ_0 and σ_0 are available for μ and σ, replace $\overline{\overline{X}}$ with μ_0 and $\hat{\sigma}$ with σ_0 in Table 39.24 and Table 39.25.

Note that the limits vary with n_i. The formulas for median limits assume that the data are normally distributed.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify μ_0 with the MU0= option or with the variable _MEAN_ in a LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in a LIMITS= data set.

Note: You can suppress the display of the control limits with the NOLIMITS option. This is useful for creating standard side-by-side box-and-whisker plots.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables can be saved:

Table 39.26. OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
_CP_	capability index C_p
CPK	capability index C_{pk}
CPL	capability index CPL
CPM	capability index C_{pm}
CPU	capability index CPU
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLM	lower control limit for subgroup median
LCLR	lower control limit for subgroup range
LCLS	lower control limit for subgroup standard deviation
LCLX	lower control limit for subgroup mean
LIMITN	nominal sample size associated with the control limits
LSL	lower specification limit
MEAN	process mean (value of central line on box chart)
_R_	value of central line on R chart
_S_	value of central line on s chart
SIGMAS	multiple (k) of standard error of $\bar{X}_i$ or M_i
STDDEV	process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*subgroup-variable* specified in the BOXCHART statement
TARGET	target value
TYPE	type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLM	upper control limit for subgroup median
UCLR	upper control limit for subgroup range
UCLS	upper control limit for subgroup standard deviation
UCLX	upper control limit for subgroup mean
USL	upper specification limit
VAR	*process* specified in the BOXCHART statement

Notes:

1. The variables _LCLM_ and _UCLM_ are included if you specify CONTROLSTAT=MEDIAN; otherwise, the variables _LCLX_ and _UCLX_ are included.
2. The variables _LCLR_, _R_, and _UCLR_ are included if you specify the RANGES option; otherwise, the variables _LCLS_, _S_, and _UCLS_ are included. These variables are not used to create box charts, but they enable the OUTLIMITS= data set to be used as a LIMITS= data set with the XRCHART, XSCHART, MRCHART, SCHART, and RCHART statements.

3. If the control limits vary with subgroup sample size, the special missing value *V* is assigned to the variables _LIMITN_, _LCLX_, _UCLX_, _LCLM_, _UCLM_, _LCLR_, _R_, _UCLR_, _LCLS_, _S_, and _UCLS_.
4. If the limits are defined in terms of a multiple k of the standard error of $\bar{X}_i$, the value of _ALPHA_ is computed as $\alpha = 2(1 - \Phi(k))$, where $\Phi(\cdot)$ is the standard normal distribution function. If the limits are defined in terms of a multiple k of the standard error of M_i, the value of _ALPHA_ is computed as $\alpha = 2(1 - F_{med}(k, n))$, where $F_{med}(\cdot, n)$ is the cumulative distribution function of the median of a random sample of n standard normally distributed observations, and n is the value of _LIMITN_. If _LIMITN_ has the special missing value *V*, this value is assigned to _ALPHA_.
5. If the limits for means are probability limits, the value of _SIGMAS_ is computed as $k = \Phi^{-1}(1 - \alpha/2)$, where Φ^{-1} is the inverse standard normal distribution function. If the limits for medians are probability limits, the value of _SIGMAS_ is computed as $k = F_{med}^{-1}(1-\alpha/2, n)$, where $F_{med}^{-1}(\cdot, n)$ is the inverse distribution function of the median of a random sample of n standard normally distributed observations, and n is the value _LIMITN_. If _LIMITN_ has the special missing value *V*, this value is assigned to _SIGMAS_.
6. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you provide a target value with the TARGET= option. See "Capability Indices" on page 1774 for computational details.
7. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the BOXCHART statement. For an example, see "Saving Control Limits" on page 1248.

OUTBOX= Data Set

The OUTBOX= data set saves subgroup summary statistics, control limits, and outlier values. The following variables can be saved:

- the *subgroup-variable*
- the variable _VAR_, containing the process variable name
- the variable _TYPE_, identifying features of box-and-whisker plots
- the variable _VALUE_, containing values of box-and-whisker plot features
- the variable _ID_, containing labels for outliers
- the variable _HTML_, containing URLs associated with box-and-whisker plot features

_ID_ is included in the OUTBOX= data set only if one of the keywords SCHEMATICID or SCHEMATICIDFAR is specified with the BOXSTYLE= option. _HTML_ is present only if one or more of the HTML=, OUTHIGHHTML=, OUTLOWHTML=, or POINTSHTML= options are specified.

Each observation in an OUTBOX= data set records the value of a single feature of one subgroup's box-and-whisker plot, such as its mean. The _TYPE_ variable identifies the feature whose value is recorded in _VALUE_. The following table lists valid _TYPE_ variable values:

Table 39.27. Valid _TYPE_ Values in an OUTBOX= Data Set

TYPE Value	Description
N	subgroup size
SIGMAS	multiple (k) of standard error of $\bar{X}_i$ or M_i
ALPHA	probability (α) of exceeding limits
LIMITN	nominal sample size associated with control limits
LCLM	lower control limit for subgroup median
LCLX	lower control limit for subgroup mean
UCLM	upper control limit for subgroup median
UCLX	upper control limit for subgroup mean
PROCMED	process median
PROCMEAN	process mean
EXLIM	control limit exceeded on box chart
TREND	trend variable value
MIN	minimum subgroup value
Q1	subgroup first quartile
MEDIAN	subgroup median
MEAN	subgroup mean
Q3	subgroup third quartile
MAX	subgroup maximum value
LOW	low outlier value
HIGH	high outlier value
LOWHISKR	low whisker value, if different from MIN
HIWHISKR	high whisker value, if different from MAX
FARLOW	low far outlier value
FARHIGH	high far outlier value

Additionally, the following variables, if specified, are included:

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables can be saved:

- the *subgroup-variable*
- a subgroup minimum variable named by the prefix *process* suffixed with *L*
- a subgroup first-quartile variable named by the prefix *process* suffixed with *1*
- a subgroup mean variable named by the prefix *process* suffixed with *X*

- a subgroup median variable named by the prefix *process* suffixed with *M*
- a subgroup third-quartile variable named by the prefix *process* suffixed with *3*
- a subgroup maximum variable named by the prefix *process* suffixed with *H*
- a subgroup sample size variable named by the prefix *process* suffixed with *N*
- a subgroup range variable named by the prefix *process* suffixed with *R* or a subgroup standard deviation variable named by *process* suffixed with *S*

A subgroup range variable is included if you specify the RANGES option; otherwise, a subgroup standard deviation variable is included.

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the BOXCHART statement. For example, consider the following statements:

```
proc shewhart data=steel;
   boxchart (width diameter)*lot / outhistory=summary;
run;
```

The data set SUMMARY contains variables named LOT, WIDTHL, WIDTH1, WIDTHM, WIDTHX, WIDTH3, WIDTHH, WIDTHS, WIDTHN, DIAMTERL, DIAMTER1, DIAMTERM, DIAMTERX, DIAMTER3, DIAMTERH, DIAMTERS, and DIAMTERN.

The variables WIDTHS and DIAMTERS are included since the RANGES option is not specified. If you specified the RANGES option, the data set SUMMARY would contain the variables WIDTHR and DIAMTERR rather than WIDTHS and DIAMTERS.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 1246.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables can be saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on box chart
LCLM	lower control limit for median
LCLX	lower control limit for mean
LIMITN	nominal sample size associated with the control limits
MEAN	process mean
SIGMAS	multiple (k) of the standard error associated with control limits
subgroup	values of the subgroup variable
SUBMAX	subgroup maximum
SUBMED	subgroup median
SUBMIN	subgroup minimum
SUBN	subgroup sample size
SUBQ1	subgroup first quartile (25^{th} percentile)
SUBQ3	subgroup third quartile (75^{th} percentile)
SUBX	subgroup mean
TESTS	tests for special causes signaled on box chart
UCLM	upper control limit for median
UCLX	upper control limit for mean
VAR	*process* specified in the BOXCHART statement

The variables _LCLM_ and _UCLM_ are included if you specify CONTROLSTAT=MEDIAN; otherwise, the variables _LCLX_ and _UCLX_ are included. In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)
- _TREND_ (if the TRENDVAR= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The k^{th} character of a value of _TESTS_ is k if Test k is positive at that subgroup. For example, if you request all eight tests and Tests 2 and 8 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, an 8 for the eighth character, and blanks for the other six characters.

3. The variables _EXLIM_ and _TESTS_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1248.

ODS Tables

The following table summarizes the ODS tables that you can request with the BOXCHART statement.

Table 39.28. ODS Tables Produced with the BOXCHART Statement

Table Name	Description	Options
BOXCHART	box plot summary statistics	TABLE, TABLEALL, TABLEBOX, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the BOXCHART statement must be a SAS variable in the data set. This variable provides measurements which must be grouped into subgroup samples indexed by the *subgroup-variable*. The *subgroup-variable*, specified in the BOXCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. If the i^{th} subgroup contains n_i measurements, there should be n_i consecutive observations for which the value of the *subgroup-variable* is the index of the i^{th} subgroup. For example, if each subgroup contains 20 items and there are 30 subgroup samples, the DATA= data set should contain 600 observations. Other variables that can be read from a DATA= data set include

- _PHASE_ (if READPHASES= is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 1936).

For an example of a DATA= data set, see "Creating Box Charts from Raw Data" on page 1240.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set CONLIMS:*

```
proc shewhart data=info limits=conlims;
   boxchart weight*batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set; see Table 39.26 on page 1269. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLX_, _MEAN_, and _UCLX_ or (if you specify CONTROLSTAT=MEDIAN) the variables _LCLM_, _MEAN_, and _UCLM_. These variables specify the control limits directly.
- the variables _MEAN_ and _STDDEV_, which are used to calculate the control limits according to the equations in Table 39.24 on page 1267 and Table 39.25 on page 1268

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are `ESTIMATE`, `STANDARD`, `STDMU`, and `STDSIGMA`.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1251.

*In Release 6.09 and in earlier releases, it is necessary to specify the READLIMITS option.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART, CUSUM, or MACONTROL procedures or to read output data sets created with SAS summarization procedures, such as PROC UNIVARIATE.

A HISTORY= data set used with the BOXCHART statement must contain the following:

- the *subgroup-variable*
- a subgroup minimum variable for each *process*
- a subgroup first-quartile variable for each *process*
- a subgroup median variable for each *process*
- a subgroup mean variable for each *process*
- a subgroup third-quartile variable for each *process*
- a subgroup maximum variable for each *process*
- a subgroup sample size variable for each *process*
- either a subgroup range variable or a subgroup standard deviation variable for each *process*

If you specify the RANGES option, the subgroup range variable must be included; otherwise, the subgroup standard deviation variable must be included.

The names of the subgroup summary statistics variables must be the *process* name concatenated with the following special suffix characters:

Subgroup Summary Statistic	Suffix Character
subgroup minimum	L
subgroup first-quartile	1
subgroup median	M
subgroup mean	X
subgroup third-quartile	3
subgroup maximum	H
subgroup sample size	N
subgroup range	R
subgroup standard deviation	S

For example, consider the following statements:

```
proc shewhart history=summary;
   boxchart (weight yldstren)*batch;
run;
```

The data set SUMMARY must include the variables BATCH, WEIGHTL, WEIGHT1, WEIGHTM, WEIGHTX, WEIGHT3, WEIGHTH, WEIGHTS, WEIGHTN, YLDSRENL, YLDSREN1, YLDSRENM, YLDSRENX, YLDSREN3, YLDSRENH, YLDSRENS, and YLDSRENN.

If the RANGES option were specified in the preceding BOXCHART statement, it would be necessary for SUMMARY to include the variables WEIGHTR and YLDSRENR rather than WEIGHTS and YLDSRENS.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if READPHASES= is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (see "Displaying Stratification in Phases" on page 1936 for an example).

For an example of a HISTORY= data set, see "Creating Box Charts from Subgroup Summary Data" page 1243.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in Chapter 56, "Specialized Control Charts."

The following table lists the variables required in a TABLE= data set used with the BOXCHART statement:

Table 39.29. Variables Required in a TABLE= Data Set

Variable	Description
LCLM	lower control limit for median
LCLX	lower control limit for mean
LIMITN	nominal sample size associated with the control limits
MEAN	process mean
subgroup-variable	values of the *subgroup-variable*
SUBMAX	subgroup maximum
SUBMIN	subgroup minimum
SUBMED	subgroup median
SUBN	subgroup sample size
SUBQ1	subgroup first quartile (25^{th} percentile)
SUBQ3	subgroup third quartile (75^{th} percentile)
SUBX	subgroup mean
UCLM	upper control limit for median
UCLX	upper control limit for mean

Note that if you specify CONTROLSTAT=MEDIAN, the variables _LCLM_, _SUBMED_, and _UCLM_ are required; otherwise, the variables _LCLX_, _SUBX_, and _UCLX_ are required.

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1248.

BOX= Data Set

You can read summary statistics, control limits, and outlier values from a BOX= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTBOX= data set created in a previous run of the SHEWHART procedure to display a box chart.

A BOX= data set must contain the following variables:

- the group variable
- _VAR_, containing the process variable name
- _TYPE_, identifying features of box-and-whisker plots
- _VALUE_, containing values of those features

Each observation in a BOX= data set records the value of a single feature of one subgroup's box-and-whisker plot, such as its mean. The _TYPE_ variable identifies the feature whose value is recorded in a given observation. The following table lists valid _TYPE_ variable values:

Table 39.30. Valid _TYPE_ Values in a BOX= Data Set

TYPE Value	Description
N	subgroup size
SIGMAS	multiple (k) of standard error of $\bar{X}_i$ or M_i
ALPHA	probability (α) of exceeding limits
LIMITN	nominal sample size associated with control limits
LCLM	lower control limit for subgroup median
LCLX	lower control limit for subgroup mean
UCLM	upper control limit for subgroup median
UCLX	upper control limit for subgroup mean
PROCMED	process median
PROCMEAN	process mean
EXLIM	control limit exceeded on box chart
TREND	trend variable value
MIN	minimum subgroup value
Q1	subgroup first quartile
MEDIAN	subgroup median
MEAN	subgroup mean
Q3	subgroup third quartile
MAX	subgroup maximum value
LOW	low outlier value
HIGH	high outlier value
LOWHISKR	low whisker value, if different from MIN
HIWHISKR	high whisker value, if different from MAX
FARLOW	low far outlier value
FARHIGH	high far outlier value

The features identified by the _TYPE_ values N, LCLM or LCLX, UCLM or UCLX, PROCMED or PROCMEAN, MIN, Q1, MEDIAN, MEAN, Q3, and MAX are required for each subgroup.

Other variables that can be read from a BOX= data set include:

- the variable _ID_, containing labels for outliers
- the variable _HTML_, containing URLs to be associated with features on box plots
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

When you specify one of the keywords SCHEMATICID or SCHEMATICIDFAR with the BOXSTYLE= option, values of _ID_ are used as outlier labels. If _ID_ does not exist in the BOX= data set, the values of the first variable listed in the ID statement are used.

Methods for Estimating the Standard Deviation

When control limits are computed from the input data, three methods (referred to as default, MVLUE and RMSDF) are available for estimating the process standard deviation σ. The method depends on whether you specify the RANGES option. If you specify this option, σ is estimated using subgroup ranges; otherwise, σ is estimated using subgroup standard deviations.

Default Method Based on Subgroup Standard Deviations

If you do not specify the RANGES option, the default estimate for σ is

$$\hat{\sigma} = \frac{s_1/c_4(n_1) + \cdots + s_N/c_4(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, s_i is the sample standard deviation of the i^{th} subgroup

$$s_i = \sqrt{\frac{1}{n_i - 1}\sum_{j=1}^{n_i}(x_{ij} - \bar{X}_i)^2}$$

and

$$c_4(n_i) = \frac{\Gamma(n_i/2)\sqrt{2/(n_i - 1)}}{\Gamma((n_i - 1)/2)}$$

Here $\Gamma(\cdot)$ denotes the gamma function, and $\bar{X}_i$ denotes the i^{th} subgroup mean. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$. If the observations are normally distributed, the expected value of s_i is $c_4(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the *ASTM Manual on Presentation of Data and Control Chart Analysis* (1976).

Default Method Based on Subgroup Ranges

If you specify the RANGES option, the default estimate for σ is

$$\hat{\sigma} = \frac{R_1/d_2(n_1) + \cdots + R_N/d_2(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, and R_i is the sample range of the observations $x_{i1}, \ldots, x_{in_i}$ in the i^{th} subgroup.

$$R_i = \max_{1 \leq j \leq n_i}(x_{ij}) - \min_{1 \leq j \leq n_i}(x_{ij})$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$. The unbiasing factor $d_2(n_i)$ is defined so that, if the observations are normally distributed, the expected value of R_i is $d_2(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the *ASTM Manual on Presentation of Data and Control Chart Analysis* (1976).

MVLUE Method Based on Subgroup Standard Deviations

If you do not specify the RANGES option and specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). This estimate is a weighted average of N unbiased estimates of σ of the form $s_i/c_4(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{h_1 s_1/c_4(n_1) + \cdots + h_N s_N/c_4(n_N)}{h_1 + \cdots + h_N}$$

where

$$h_i = \frac{[c_4(n_i)]^2}{1 - [c_4(n_i)]^2}$$

A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

MVLUE Method Based on Subgroup Ranges

If you specify the RANGES option and SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). The MVLUE is a weighted average of N unbiased estimates of σ of the form $R_i/d_2(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{f_1 R_1/d_2(n_1) + \cdots + f_N R_N/d_2(n_N)}{f_1 + \cdots + f_N}$$

where

$$f_i = \frac{[d_2(n_i)]^2}{[d_3(n_i)]^2}$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The unbiasing factor $d_3(n_i)$ is defined so that, if the observations are normally distributed, the expected value of σ_{R_i} is $d_3(n_i)\sigma$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

RMSDF Method Based on Subgroup Standard Deviations

If you do not specify the RANGES option and specify SMETHOD=RMSDF, a weighted root-mean-square estimate is computed for σ.

$$\hat{\sigma} = \frac{\sqrt{(n_1 - 1)s_1^2 + \cdots + (n_N - 1)s_N^2}}{c_4(n)\sqrt{n_1 + \cdots + n_N - N}}$$

The weights are the degrees of freedom $n_i - 1$. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$.

If the unknown standard deviation σ is constant across subgroups, the root-mean-square estimate is more efficient than the minimum variance linear unbiased estimate. However, in process control applications, it is generally not assumed that σ is constant, and if σ varies across subgroups, the root-mean-square estimate tends to be more inflated than the MVLUE.

Percentile Definitions

You can use the PCTLDEF= option to specify one of five definitions for computing quantile statistics (percentiles). Let n equal the number of nonmissing values for a variable, and let $x_1, x_2, \ldots, x_n$ represent the ordered values of the process variable. For the t^{th} percentile, set $p = t/100$, and express np as

$$np = j + g$$

where j is the integer part of np, and g is the fractional part of np.

The t^{th} percentile (call it y) can be defined in five ways, as described in the next five sections.

PCTLDEF=1

This uses the weighted average at x_{np}

$$y = (1 - g)x_j + gx_{j+1}$$

where x_0 is taken to be x_1.

PCTLDEF=2

This uses the observation numbered closest to np

$$y = x_i$$

where i is the integer part of $np + 1/2$.

PCTLDEF=3

This uses the empirical distribution function

$$\begin{array}{ll} y = x_j & \text{if } g = 0 \\ y = x_{j+1} & \text{if } g > 0 \end{array}$$

PCTLDEF=4

This uses the weighted average aimed at $x_{p(n+1)}$

$$y = (1 - g)x_j + gx_{j+1}$$

where $(n + 1)p = j + g$, and where x_{n+1} is taken to be x_n.

PCTLDEF=5

This uses the empirical distribution function with averaging

$$y = (x_j + x_{j+1})/2 \quad \text{if } g = 0$$
$$y = x_{j+1} \quad \text{if } g > 0$$

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical (box chart)	DATA=	*process*
Vertical (box chart)	HISTORY=	subgroup mean variable
Vertical (box chart)	TABLE=	_SUBX_

Note that if you specify the CONTROLSTAT=MEDIAN option, you should assign the label to the subgroup median variable in a HISTORY= data set or to the variable _SUBMED_ in an TABLE= data set.

If you specify the TRENDVAR= option, you can provide distinct labels for the vertical axes of the box and trend charts by breaking the vertical axis into two parts with a split character. Specify the split character with the SPLIT= option. The first part labels the vertical axis of the box chart, and the second part labels the vertical axis of the trend chart.

For an example, see "Labeling Axes" on page 1966.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples

This section provides advanced examples of the BOXCHART statement.

Example 39.1. Using Box Charts to Compare Subgroups

See SHWBOX4 in the SAS/QC Sample Library

In this example, a box chart is used to compare the delay times for airline flights during the Christmas holidays with the delay times prior to the holiday period. The following statements create a data set named TIMES with the delay times in minutes for 25 flights each day. When a flight is cancelled, the delay is recorded as a missing value.

```
data times;
   informat day date7. ;
   format   day date7. ;
   input day @ ;
   do flight=1 to 25;
      input delay @ ;
      output;
      end;
datalines;
16DEC88   4  12   2   2  18   5   6  21   0   0
          0  14   3   .   2   3   5   0   6  19
          7   4   9   5  10
17DEC88   1  10   3   3   0   1   5   0   .   .
          1   5   7   1   7   2   2  16   2   1
          3   1  31   5   0
18DEC88   7   8   4   2   3   2   7   6  11   3
          2   7   0   1  10   2   3  12   8   6
          2   7   2   4   5
19DEC88  15   6   9   0  15   7   1   1   0   2
          5   6   5  14   7  20   8   1  14   3
         10   0   1  11   7
20DEC88   2   1   0   4   4   6   2   2   1   4
          1  11   .   1   0   6   5   5   4   2
          2   6   6   4   0
21DEC88   2   6   6   2   7   7   5   2   5   0
          9   2   4   2   5   1   4   7   5   6
          5   0   4  36  28
22DEC88   3   7  22   1  11  11  39  46   7  33
         19  21   1   3  43  23   9   0  17  35
         50   0   2   1   0
23DEC88   6  11   8  35  36  19  21   .   .   4
          6  63  35   3  12  34   9   0  46   0
          0  36   3   0  14
24DEC88  13   2  10   4   5  22  21  44  66  13
          8   3   4  27   2  12  17  22  19  36
          9  72   2   4   4
25DEC88   4  33  35   0  11  11  10  28  34   3
         24   6  17   0   8   5   7  19   9   7
         21  17  17   2   6
26DEC88   3   8   8   2   7   7   8   2   5   9
```

```
          2   8   2  10  16   9   5  14  15   1
         12   2   2  14  18
;
run;
```

First, the MEANS procedure is used to count the number of cancelled flights for each day. This information is then added to the data set TIMES.

```
proc means data=times noprint;
   var delay;
   by day ;
   output out=cancel nmiss=ncancel;

data times;
   merge times cancel;
   by day;
run;
```

The following statements create a data set named WEATHER that contains information about possible causes for delays. This data set is merged with the data set TIMES.

```
data weather;
   informat day date7. ;
   format   day date7. ;
   length reason $ 16 ;
input day flight reason & ;
datalines;
16DEC88  8   Fog
17DEC88  18  Snow Storm
17DEC88  23  Sleet
21DEC88  24  Rain
21DEC88  25  Rain
22DEC88  7   Mechanical
22DEC88  15  Late Arrival
24DEC88  9   Late Arrival
24DEC88  22  Late Arrival
;
run;

data times;
   merge times weather;
   by day flight;
run;
```

Next, control limits are established using the delays prior to the holiday period.

```
   proc shewhart data=times;
      where day <= '21DEC88'D;
      boxchart delay * day /
         nochart
         outlimits=timelim;
   run;
```

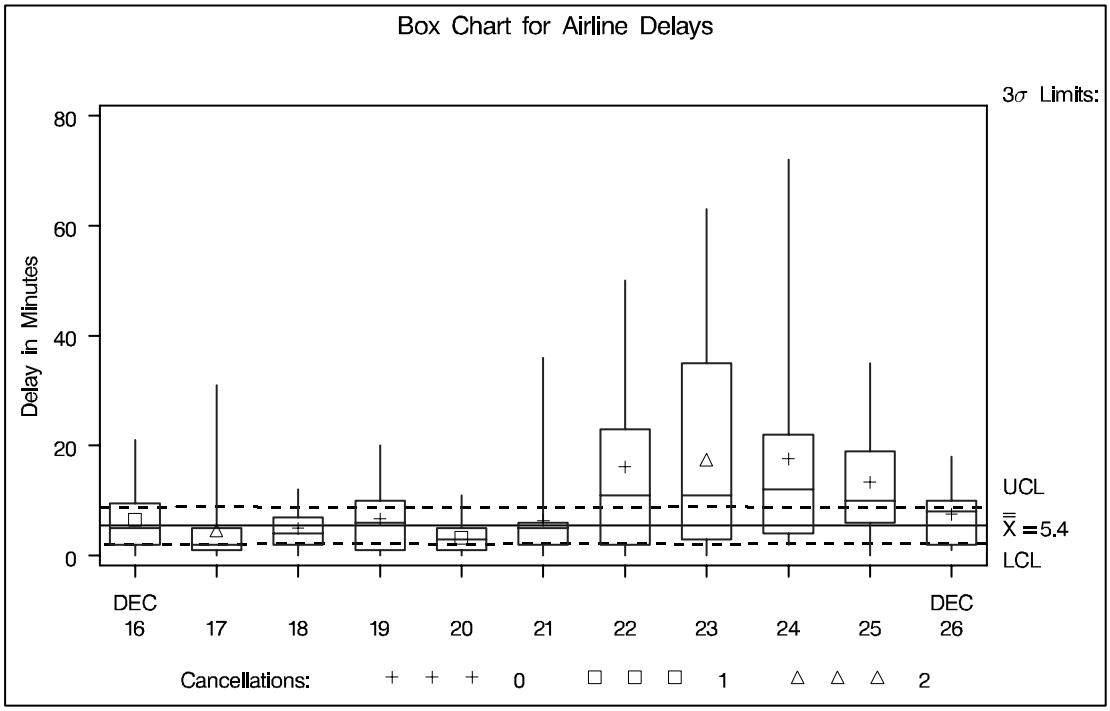
The OUTLIMITS= option names a data set (TIMELIM) that saves the control limits. The NOCHART option suppresses the display of the chart.

The following statements create a box chart for the complete set of data using the control limits in TIMELIM:

```
symbol1 v=plus     c=black;
symbol2 v=square   c=black;
symbol3 v=triangle c=black;
title 'Box Chart for Airline Delays';
proc shewhart data=times limits=timelim ;
   boxchart delay * day = ncancel /
      llimits      = 20
      readlimits
      nohlabel
      nolegend
      symbollegend = legend1;
   legend1 label=('Cancellations:');
   label delay = 'Delay in Minutes';
run;
```

The box chart is shown in Output 39.1.1. The level of the *symbol-variable* NCANCEL determines the symbol marker for each subgroup mean, and the SYMBOLLEGEND= option controls the appearance of the legend for the symbols. The NOHLABEL option suppresses the label for the horizontal axis, and the NOLEGEND option suppresses the default legend for subgroup sample sizes.

Output 39.1.1. Box Chart for Airline Data



The delay distributions from December 22 through December 25 are drastically different from the delay distributions during the pre-holiday period. Both the mean delay and the variability of the delays are much greater during the holiday period.

Example 39.2. Creating Various Styles of Box-and-Whisker Plots

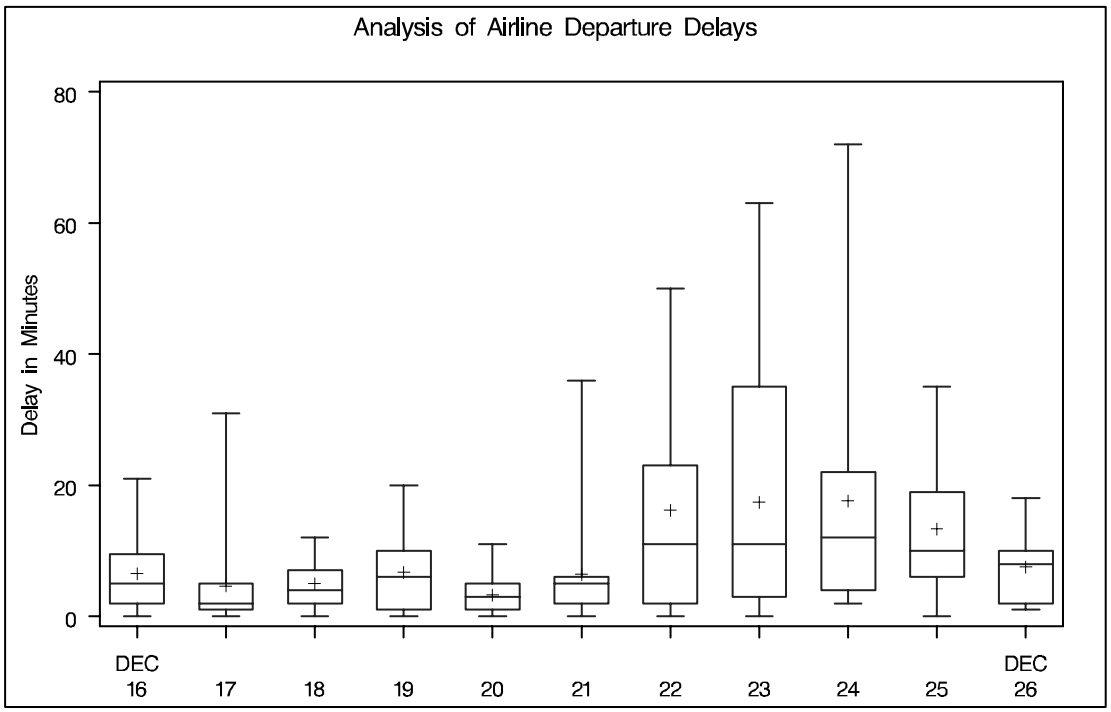
This example uses the flight delay data of the preceding example to illustrate how you can create box charts with various styles of box-and-whisker plots. For simplicity, the control limits are suppressed. The following statements create a chart, shown in Output 39.2.1, that displays *skeletal box-and-whisker plots*:

See SHWBOX5 in the SAS/QC Sample Library

```
symbol1 v=plus c=black;
title 'Analysis of Airline Departure Delays';
proc shewhart data=times limits=timelim ;
   boxchart delay * day /
      boxstyle = skeletal
      serifs
      nolimits
      nohlabel
      nolegend;
   label delay = 'Delay in Minutes';
run;
```

In a skeletal box-and-whisker plot, the whiskers are drawn from the quartiles to the extreme values of the subgroup sample. You can also request this style by omitting the BOXSTYLE= option, since this style is the default. The SERIFS option adds serifs to the whiskers (by default, serifs are omitted with the skeletal style). The NOLIMITS option suppresses the display of the control limits.

Output 39.2.1. BOXSTYLE=SKELETAL with Serifs

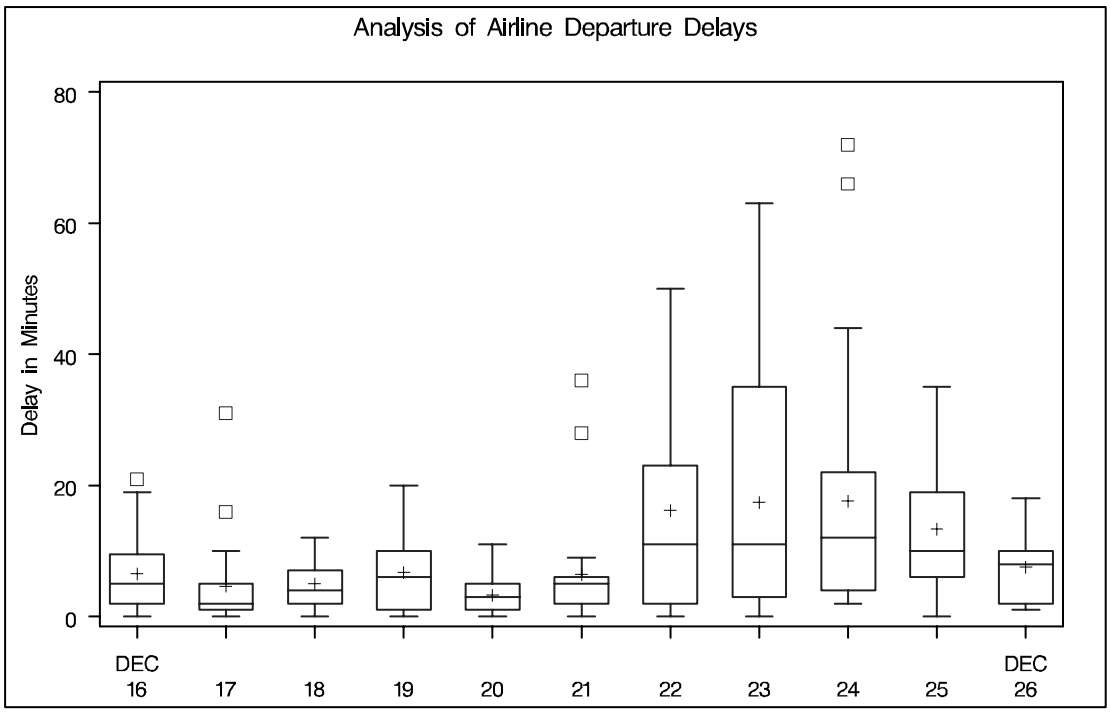


The following statements request a box chart with *schematic box-and-whisker plots*:

```
symbol1 v=plus c=black;
title 'Analysis of Airline Departure Delays';
proc shewhart data=times limits=timelim ;
   boxchart delay * day /
      boxstyle = schematic
      nolimits
      nohlabel
      nolegend;
   label delay = 'Delay in Minutes';
run;
```

The chart is shown in Output 39.2.2. When BOXSTYLE=SCHEMATIC is specified, the whiskers are drawn to the most extreme points in the subgroup sample that lie within so-called "fences." The *upper fence* is defined as the third quartile (represented by the upper edge of the box) plus 1.5 times the interquartile range (IQR). The *lower fence* is defined as the first quartile (represented by the lower edge of the box) minus 1.5 times the interquartile range. Observations outside the fences are identified with a special symbol. The default symbol is a square, and you can specify the shape and color for this symbol with the IDSYMBOL= and IDCOLOR= options. Serifs are added to the whiskers by default. For further details, see the entry for BOXSTYLE= on page 1858 in Chapter 53, "Dictionary of Options."

Output 39.2.2. BOXSTYLE=SCHEMATIC

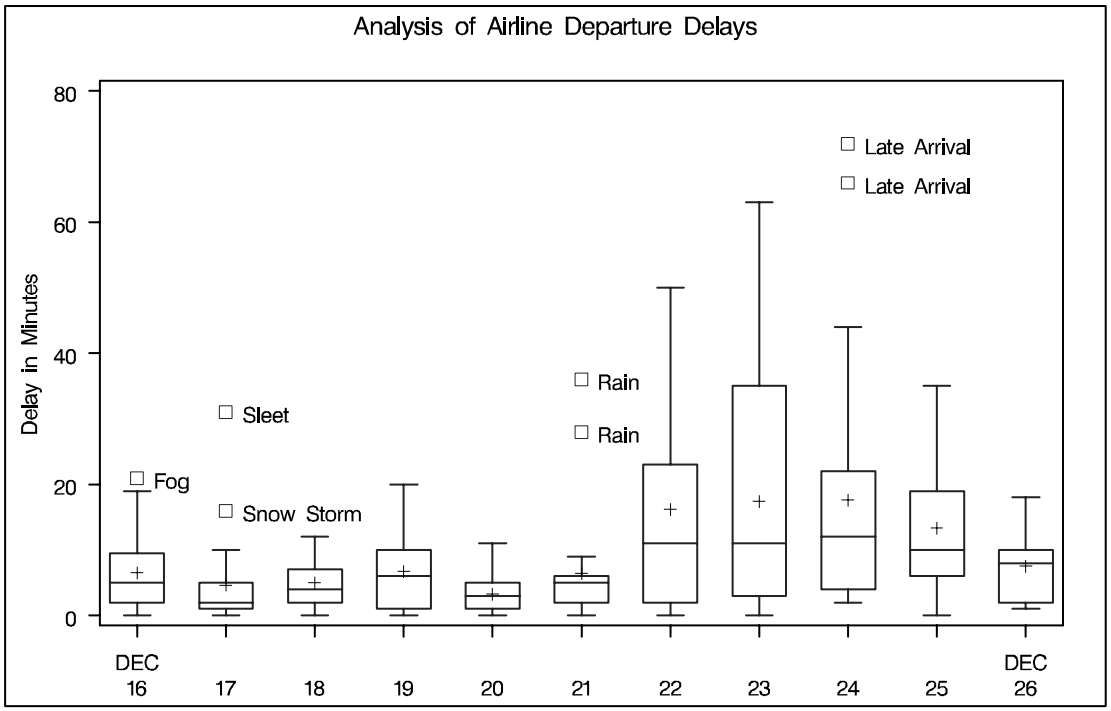


The following statements create a box chart with schematic box-and-whisker plots in which the observations outside the fences are labeled:

```
symbol1 v=plus c=black;
title 'Analysis of Airline Departure Delays';
proc shewhart data=times limits=timelim ;
   boxchart delay * day /
      boxstyle = schematicid
      llimits  = 20
      nolimits
      nohlabel
      nolegend;
   id reason;
   label delay = 'Delay in Minutes';
run;
```

The chart is shown in Output 39.2.3. If you specify BOXSTYLE=SCHEMATICID, schematic box-and-whisker plots are displayed in which the value of the first ID variable (in this case, REASON) is used to label each observation outside the fences.

Output 39.2.3. BOXSTYLE=SCHEMATICID

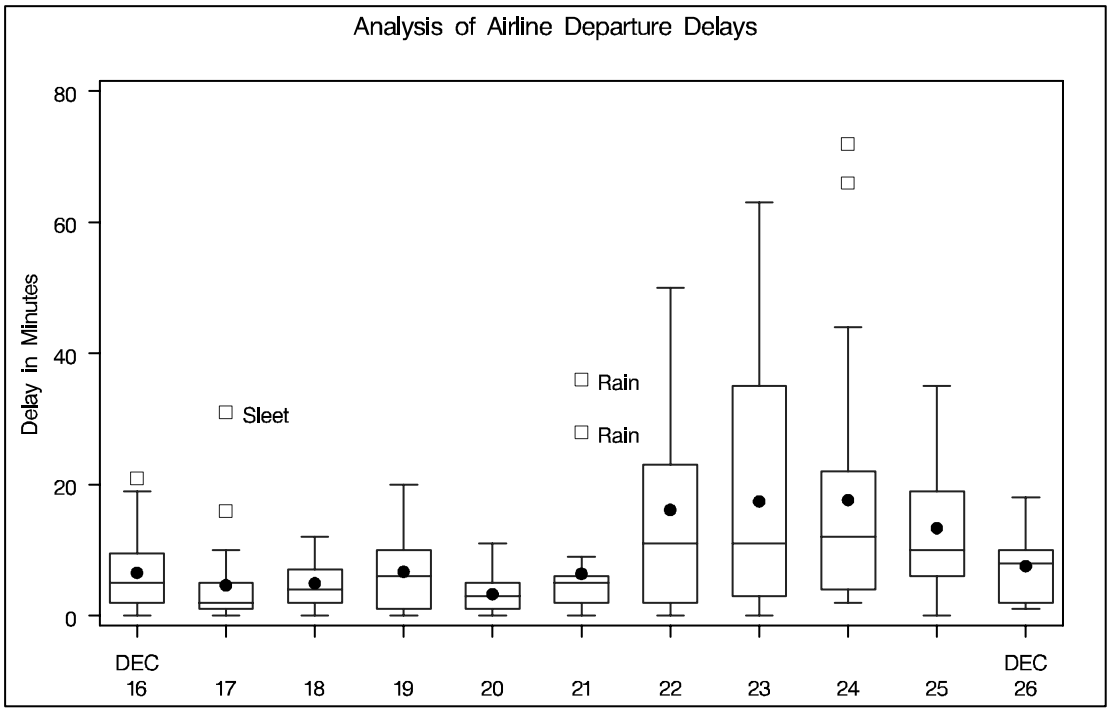


The following statements create a box chart with schematic box-and-whisker plots in which only the extreme observations outside the fences are labeled:

```
title 'Analysis of Airline Departure Delays';
proc shewhart data=times limits=timelim ;
   boxchart delay * day /
      boxstyle = schematicidfar
      nolimits
      nohlabel
      nolegend;
   id reason;
   label delay = 'Delay in Minutes';
run;
```

The chart is shown in Output 39.2.4. If you specify BOXSTYLE=SCHEMATICIDFAR, schematic box-and-whisker plots are displayed in which the value of the first ID variable is used to label each observation outside the *lower* and *upper far fences*. The *lower* and *upper far fences* are located $3\times$IQR below the 25^{th} percentile and above the 75^{th} percentile, respectively. Observations between the fences and the far fences are identified with a symbol but are not labeled.

Output 39.2.4. BOXSTYLE=SCHEMATICIDFAR



Other options for controlling the display of box-and-whisker plots include the BOXWIDTH=, BOXWIDTHSCALE=, CBOXES=, CBOXFILL=, and LBOXES= options. For details, see the corresponding entries in Chapter 53, "Dictionary of Options."

Example 39.3. Creating Notched Box-and-Whisker Plots

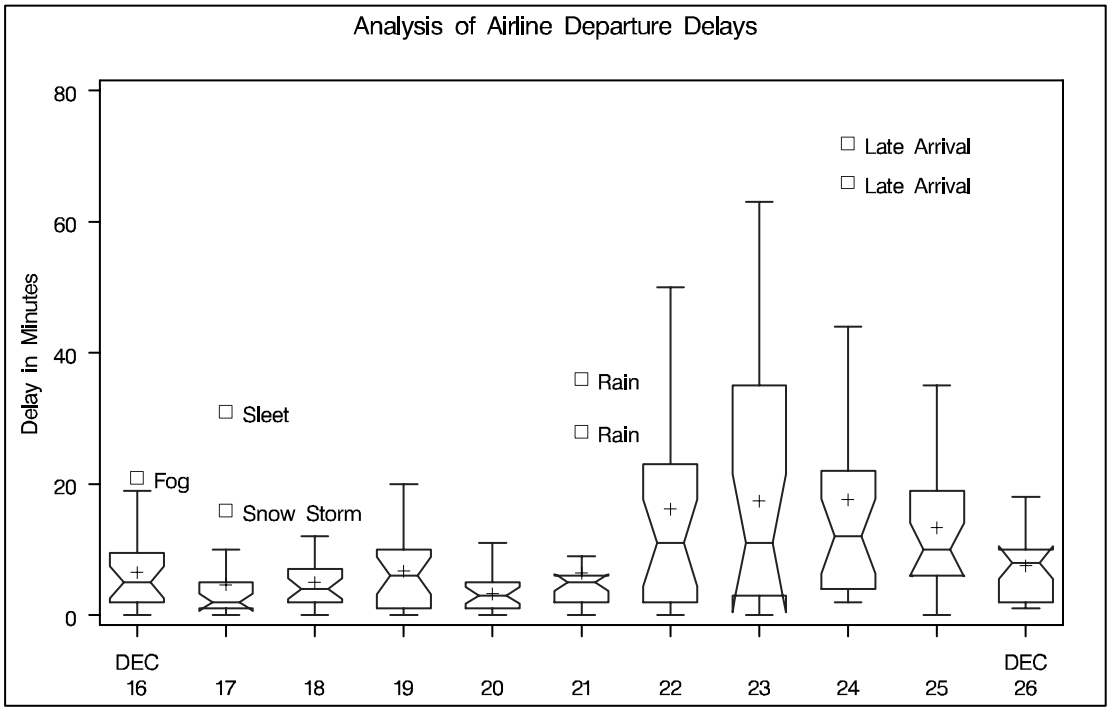
The following statements use the flight delay data of Example 39.1 to illustrate how to create side-by-side box-and-whisker plots with notches:

See SHWBOX4 in the SAS/QC Sample Library

```
symbol1 v=plus c=black;
title 'Analysis of Airline Departure Delays';
proc shewhart data=times limits=timelim ;
   boxchart delay * day /
      boxstyle = schematicid
      llimits  = 20
      nolimits
      nohlabel
      nolegend
      notches;
   id reason;
   label delay = 'Delay in Minutes';
run;
```

The control limits are suppressed with the NOLIMITS option. The notches, requested with the NOTCHES option, measure the significance of the difference between two medians. The medians are significantly different at approximately the 95% level if the notches do not overlap. For details, see the entry for NOTCHES on page 1886 in Chapter 53, "Dictionary of Options."

Output 39.3.1. Notched Side-by-Side Box-and-Whisker Plots



Example 39.4. Creating Box-and-Whisker Plots with Varying Widths

See SHWBOX7 in the SAS/QC Sample Library

This example shows how to create a box chart with box-and-whisker plots whose widths vary proportionately with the subgroup sample size. The following statements create a SAS data set named TIMES2 that contains flight departure delays (in minutes) recorded daily for eight consecutive days:

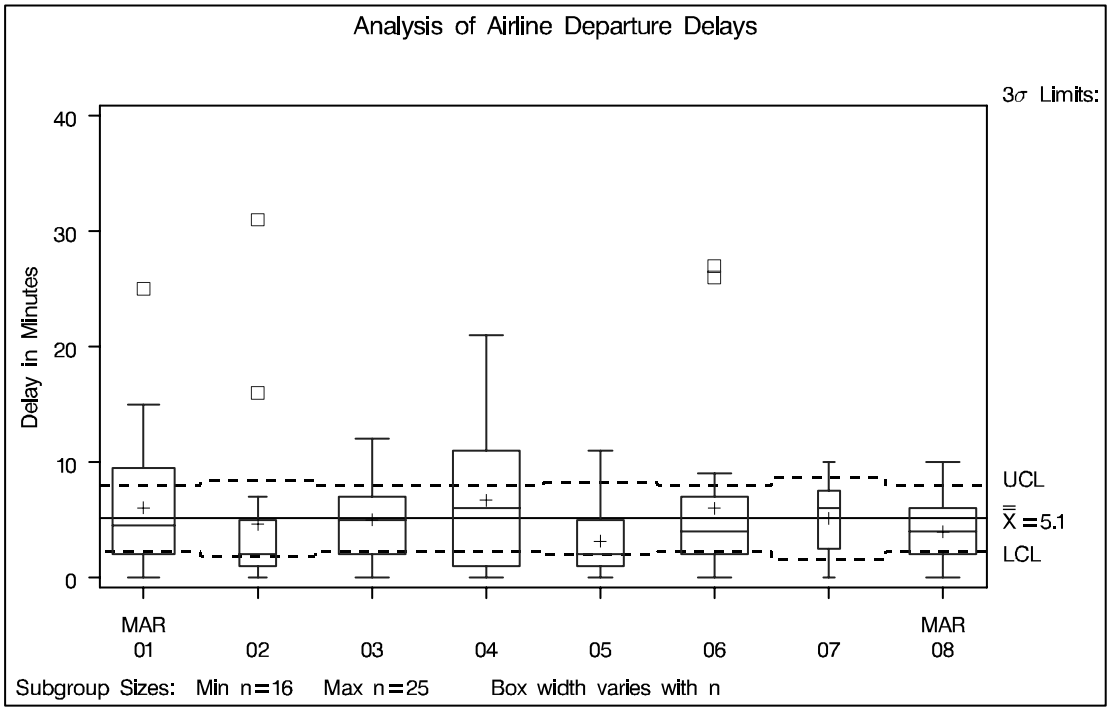
```
data times2;
   label delay = 'Delay in Minutes';
   informat day date7. ;
   format   day date7. ;
   input day @ ;
   do flight=1 to 25;
      input delay @ ;
      output;
      end;
datalines;
01MAR90   12  4   2   2  15   8   0  11   0   0
          0  12   3   .   2   3   5   0   6  25
          7   4   9   5  10
02MAR90   1   .   3   .   0   1   5   0   .   .
          1   5   7   .   7   2   2  16   2   1
          3   1  31   .   0
03MAR90   6   8   4   2   3   2   7   6  11   3
          2   7   0   1  10   2   5  12   8   6
          2   7   2   4   5
04MAR90  12   6   9   0  15   7   1   1   0   2
          5   6   5  14   7  21   8   1  14   3
         11   0   1  11   7
05MAR90   2   1   0   4   .   6   2   2   1   4
          1  11   .   1   0   .   5   5   .   2
          3   6   6   4   0
06MAR90   8   6   5   2   9   7   4   2   5   1
          2   2   4   2   5   1   3   9   7   8
          1   0   4  26  27
07MAR90   9   6   6   2   7   8   .   .  10   8
          0   2   4   3   .   .   .   7   .   6
          4   0   .   .   .
08MAR90   1   6   6   2   8   8   5   3   5   0
          8   2   4   2   5   1   6   4   5  10
          2   0   4   1   1
run;
```

The following statements create the box chart shown in Output 39.4.1:

```
title 'Analysis of Airline Departure Delays';
symbol1 v=plus c=black;
proc shewhart data=times2;
   boxchart delay * day /
      llimits       = 20
      nohlabel
      boxstyle      = schematic
      boxwidthscale = 1 ;
run;
```

The BOXWIDTHSCALE=1 option specifies that the widths of the box-and-whisker plots are to vary proportionately to the subgroup sample size n. This option is useful in situations where the sample size varies widely across subgroups. For further details, see the entry for BOXWIDTHSCALE= on page 1861 in Chapter 53, "Dictionary of Options."

Output 39.4.1. Box Chart with Box-and-Whisker Plots of Varying Widths



Example 39.5. Creating Box-and-Whisker Plots with Different Line Styles and Colors

See SHWBOX7 in the SAS/QC Sample Library

The control limits in Output 39.4.1 apply to the subgroup means. This example illustrates how you can modify the chart to indicate whether the variability of the process is in control. The following statements create a box chart for DELAY in which a dashed outline and a light gray fill color are used for a box-and-whisker plot if the corresponding subgroup standard deviation exceeds its 3σ limits.

First, the SHEWHART procedure is used to create an OUTTABLE= data set (DELAYTAB) that contains a variable (_EXLIMS_) that records which standard deviations exceed their 3σ limits.

```
proc shewhart data=times2;
   xschart delay * day / nochart
                         outtable = delaytab;
run;
```

Then, this information is used to set the line styles and fill colors as follows:

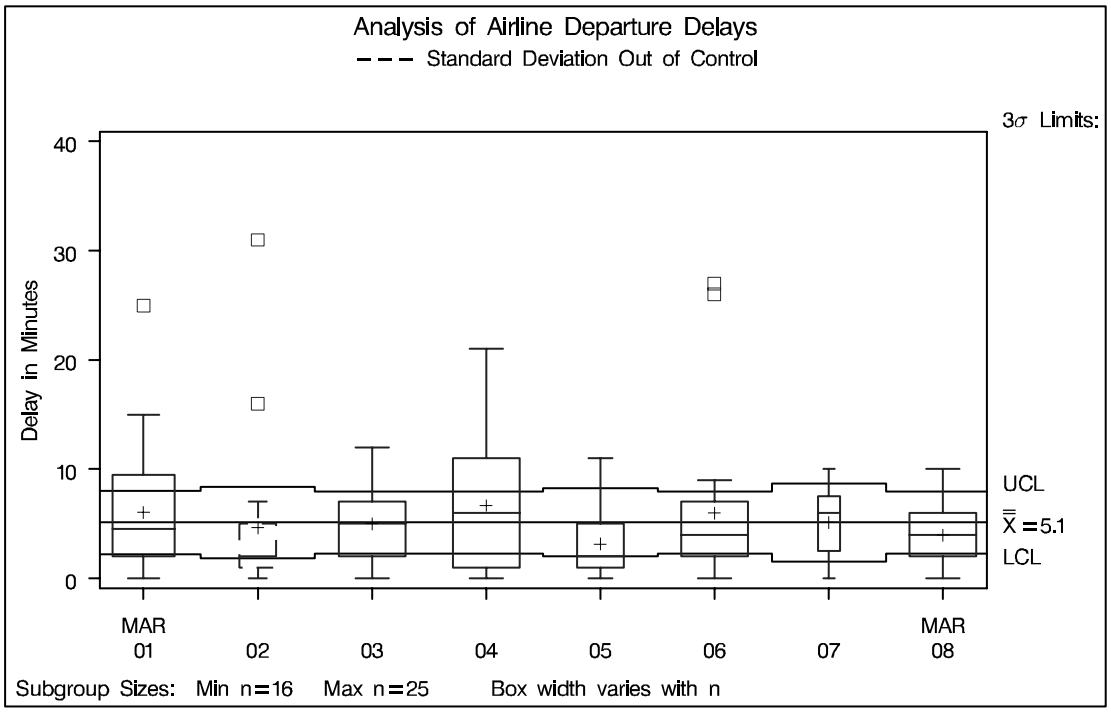
```
data delaytab;
   length boxcolor $ 8 ;
   set delaytab;
   keep day lnstyle boxcolor;
   if _exlims_ = 'UPPER' or _exlims_ = 'LOWER' then do;
      lnstyle  = 20;
      boxcolor = 'bigb' ;
      end;
   else do;
      lnstyle  = 1;
      boxcolor = 'ywh' ;
      end;
run;

data times2;
   merge times2 delaytab;
   by day;
run;
```

The following statements create the modified box chart:

```
title  'Analysis of Airline Departure Delays' ;
title2 '--- Standard Deviation Out of Control';
symbol1 v=plus c=black;
proc shewhart data=times2;
   boxchart delay * day /
        llimits       = 1
        nohlabel
        boxstyle      = schematic
        lboxes        = ( lnstyle  )
        boxwidthscale = 1 ;
run;
```

The chart is shown in Output 39.5.1. The values of the variable LNSTYLE specified with the LBOXES= option determine the outline styles for the box-and-whisker plots. The values of the variable BOXCOL specified with the CBOXFILL= option determines the fill colors. For further details, see the entries for these options in Chapter 53, "Dictionary of Options." The chart indicates that the large variability for March 2 should be checked.

Output 39.5.1. Box Chart Displaying Out-of-Control Subgroup Standard Deviations



Example 39.6. Computing the Control Limits for Subgroup Maximums

See SHWBOX3 in the SAS/QC Sample Library

This example illustrates how to compute and display control limits for the *maximum* of a subgroup sample. Subgroup samples of 20 metal braces are collected daily, and the lengths of the braces are measured in centimeters. These data are analyzed extensively in Example 51.3 on page 1828. The box chart for LOGLENG (the log of length) shown in Output 51.3.3 on page 1831 indicates that the subgroup mean is in control and that the subgroup distributions of LOGLENG are approximately normal. The following statements save the control limits for the mean of the LOGLENG in a data set named LOGLLIMS:

```
data lengdata;
   set lengdata;
   logleng=log(length-105);
run;

proc shewhart data=lengdata;
   xchart logleng*day /
      nochart
      outlimits=logllims;
run;
```

The next statements replace the control limits for the mean of LOGLENG with control limits for the maximum of LOGLENG:

```
data maxlim;
   set lengdata;
   set log1lims;
   drop expmax stdmax;
   label _lclx_ = 'Lower Limit for Maximum of 20'
         _uclx_ = 'Upper Limit for Maximum of 20'
         _mean_ = 'Central Line for Maximum of 20';
   expmax = _stddev_*1.86748 + _mean_;
   stdmax = _stddev_*0.52507;
   _lclx_ = expmax - _sigmas_*stdmax;
   _uclx_ = expmax + _sigmas_*stdmax;
   _mean_ = expmax;
   call symput('avgmax',left(put(expmax,8.1)));
run;
```

The control limits are computed using the fact that the maximum of a sample of size 20 from a normal population with zero mean and unit standard deviation has an expected value of 1.86747 and a standard deviation of 0.52509; refer to Teichroew (1956) and see Table 39.31 on page 1297. Finally, the following statements create a box chart for LOGLENG that displays control limits for the subgroup maximum:

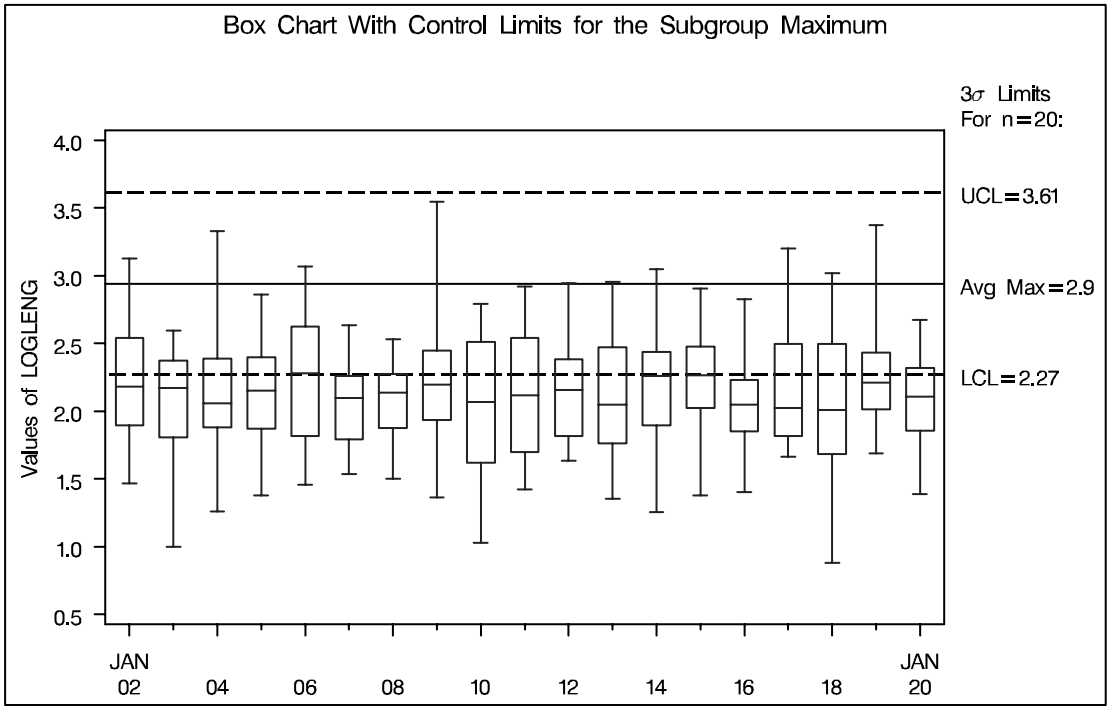
```
title 'Box Chart With Control Limits for the Subgroup Maximum';
symbol v=none;
proc shewhart data=lengdata limits=maxlim;
   boxchart logleng*day /
      ranges
      serifs
      nohlabel
      nolegend
      xsymbol="Avg Max=&AVGMAX" ;
   label logleng='Values of LOGLENG';
run;
```

The box chart, shown in Output 39.6.1, indicates that the maximum is in control since the tips of the upper whiskers fall within the control limits.

The SYMPUT call is used to pass the value of _MEAN_ in a macro variable to the SHEWHART procedure so that this value can be used to label the central line.

You can apply the variable replacement method shown here to data with sample sizes other than 20 by replacing the constants 1.86747 and 0.52509 with the appropriate values from Table 39.31. Austin (1973) describes a method for approximating these values. You can also use the preceding statements to display control limits for the subgroup minimum by changing the sign of the expected values in Table 39.31.

Output 39.6.1. Box Chart for Subgroup Maximum



The variable replacement method can also be used to create a variety of box charts, including the modifications suggested by Iglewicz and Hoaglin (1987) and Rocke (1989).

Table 39.31. Expected Values and Standard Deviations of Maximum of a Normal Sample

n	Expected Value	Standard Deviation
2	0.56418	0.82565
3	0.84628	0.74798
4	1.02937	0.70123
5	1.16296	0.66899
6	1.26720	0.64494
7	1.35217	0.62605
8	1.42360	0.61065
9	1.48501	0.59780
10	1.53875	0.58681
11	1.58643	0.57730
12	1.62922	0.56891
13	1.66799	0.56144
14	1.70338	0.55474
15	1.73591	0.54869
16	1.76599	0.54316
17	1.79394	0.53809
18	1.82003	0.53342
19	1.84448	0.52910
20	1.86747	0.52509

Example 39.7. Constructing Multi-Vari Charts

"Multi-vari" charts* are used in a variety of industries to analyze process data with nested (hierarchical) patterns of variation

- within-sample variation (for example, position within wafer)
- sample-to-sample variation within batches of samples (for example, wafer within lot)
- batch-to-batch variation (for example, across lots)

This example illustrates the construction of a "multi-vari" display. The following statements create a SAS data set named PARM that contains the value of a measured parameter (MEASURE) recorded at each of five positions on wafers produced in lots.

```
data parm;
   length _phase_ $ 5 wafer $ 2 position $ 1;
   input  _phase_ $  &  wafer $ &  position $  measure ;
   datalines;
   Lot A      01         L        2.42435
   Lot A      01         B        2.44150
   Lot A      01         C        2.42143
   Lot A      01         T        2.44960
   Lot A      01         R        2.50050
   Lot A      02         L        2.68188
   Lot A      02         B        2.57195
   Lot A      02         C        2.54678
   Lot A      02         T        2.65978
   Lot A      02         R        2.69208
   Lot A      03         L        2.18005
   Lot A      03         B        2.13593
   Lot A      03         C        2.44303
   Lot A      03         T        2.29052
   Lot A      03         R        2.25963
   Lot B      01         L        2.46573
   Lot B      01         B        2.44898
   Lot B      01         C        2.52365
   Lot B      01         T        2.74458
   Lot B      01         R        2.88328
   Lot B      02         L        2.37283
   Lot B      02         B        2.14528
   Lot B      02         C        2.53138
   Lot B      02         T        2.47408
   Lot B      02         R        2.56205

  ...

   Lot G      03         R        2.84378
run;
```

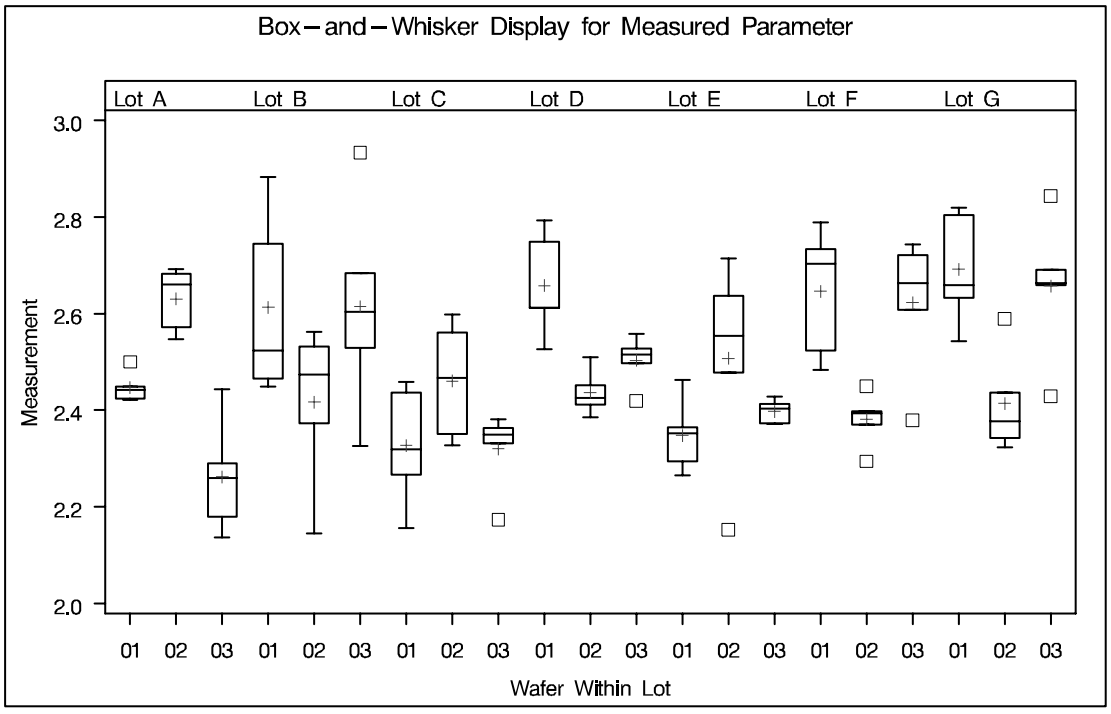
The following statements create an ordinary side-by-side box-and-whisker display for the measurements.

*Multi-vari charts should not be confused with multivariate control charts , which are discussed on page 2033.

```
symbol v=plus c=bib;
title 'Box-and-Whisker Display for Measured Parameter';
proc shewhart data=parm;
   boxchart measure*wafer /
         nolimits
         cboxes     = black
         cinfill    = ligr
         boxstyle   = schematic
         idsymbol   = square
         readphase  = all
         phaselegend
         nolegend;
label measure = 'Measurement'
      wafer   = 'Wafer Within Lot';
run;
```

The display is shown in Output 39.7.1. Here, the *subgroup-variable* is WAFER, and the option BOXSTYLE=SCHEMATIC is specified to request schematic box-and-whisker plots for the measurements in each subgroup (wafer) sample. The lot values are provided as the values of the special variable _PHASE_, which is read when the option READPHASE=ALL is specified. The option PHASELEGEND requests the legend for phase (lot) values at the top of the chart, and the NOLEGEND option suppresses the default legend for sample sizes. The NOLIMITS option suppresses the display of control limits. This option is recommended whenever you are using the BOXCHART statement to create side-by-side box-and-whisker plots.

Output 39.7.1. Box-and-Whisker Plot Using BOXSTYLE=SCHEMATIC

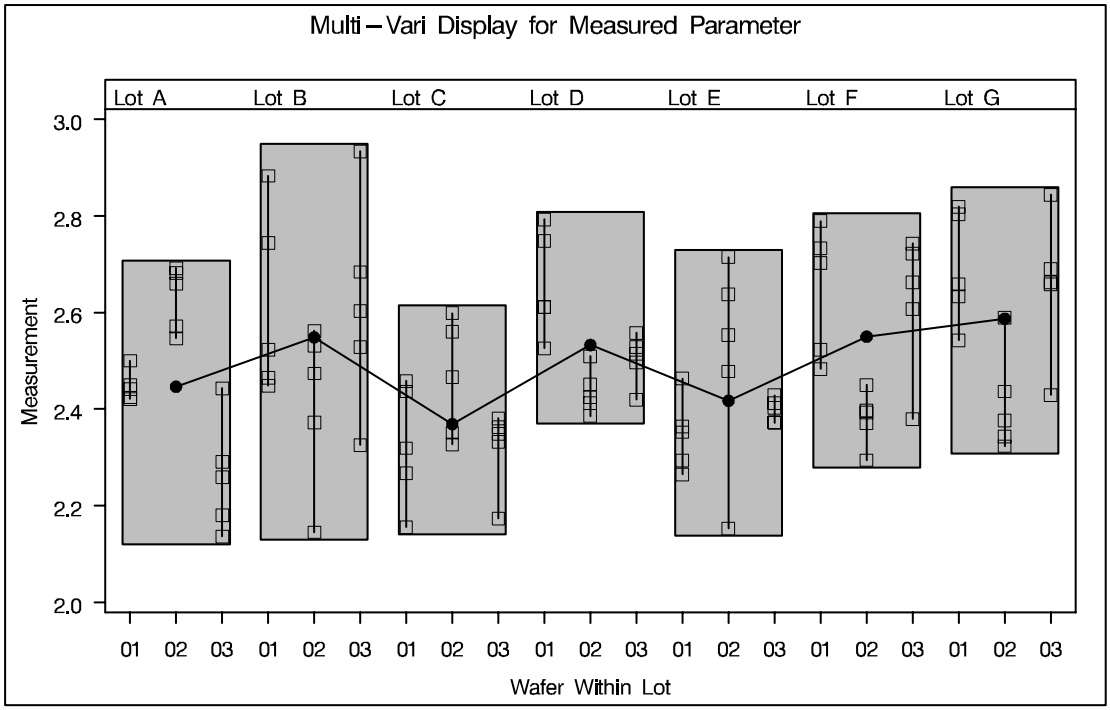


The box-and-whisker display in Output 39.7.1 is not particularly appropriate for these data since there are only five measurements in each wafer and since the variation within each wafer may depend on the position, which is not indicated. The next statements use the BOXCHART statement to produce a multi-vari chart for the same data.

```
symbol v=none;
title 'Multi-Vari Display for Measured Parameter';
proc shewhart data=parm;
   boxchart measure*wafer /
      nolimits
      boxstyle           = pointsjoin
      cboxes             = black
      idsymbol           = square
      cphaseboxfill      = ligr
      cphasebox          = black
      cphasemeanconnect  = black
      phasemeansymbol    = dot
      readphase          = all
      phaselegend
      nolegend;
label measure = 'Measurement'
      wafer   = 'Wafer Within Lot';
run;
```

The display is shown in Output 39.7.2.

Output 39.7.2. Multi-Vari Chart Using BOXSTYLE=POINTSJOIN

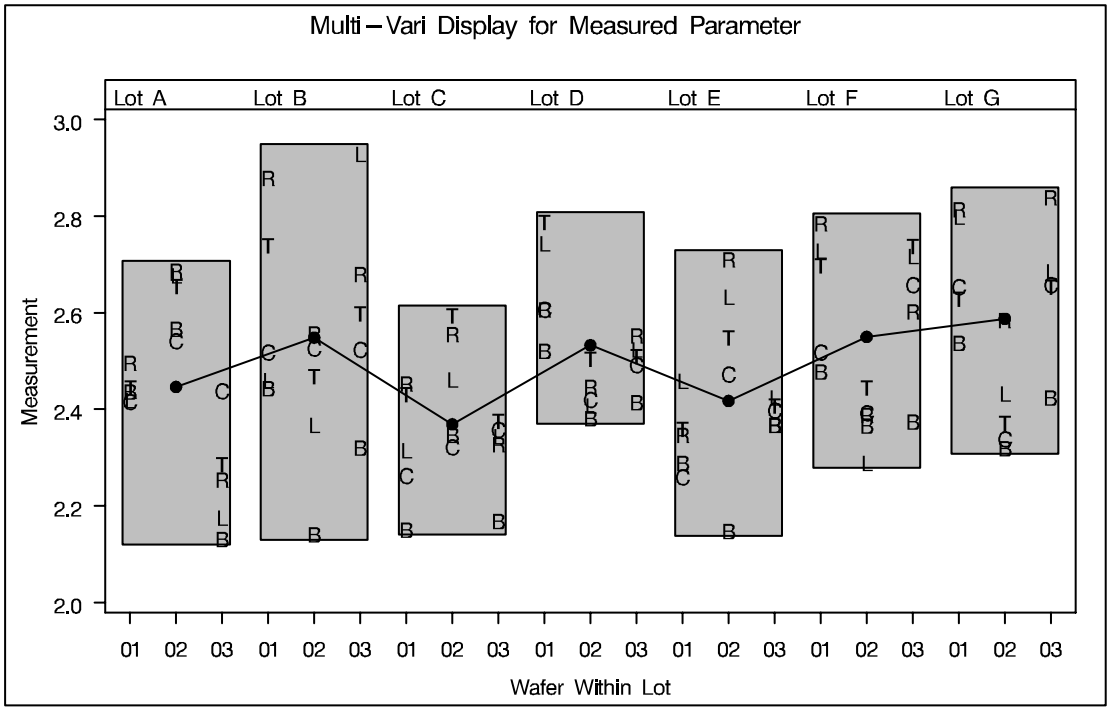


The option BOXSTYLE=POINTSJOIN specifies that the values for each wafer are to be displayed as points joined by a vertical line. The IDSYMBOL= option specifies the symbol marker for the points. The option V=NONE in the SYMBOL statement is specified to suppress the symbol for the wafer averages shown in Output 39.7.1. The option CPHASEBOX=BLACK specifies that the points for each lot are to be enclosed in a black box, and the CPHASEBOXFILL= option specifies the fill color for the box. The option CPHASEMEANCONNECT=BLACK specifies that the means of the lots are to be connected with black lines, and the PHASEMEANSYMBOL= option specifies the symbol marker for the lot means.

The following statements create a slightly different multi-vari chart using the values of the variable POSITION to identify the measurements for each wafer. Note that the option BOXSTYLE=POINTSID is specified and that POSITION is specified as the ID variable. The display is shown in Output 39.7.3.

```
symbol v=none;
title 'Multi-Vari Display for Measured Parameter';
proc shewhart data=parm;
   boxchart measure*wafer /
      nolimits
      cboxes            = black
      cphaseboxfill     = ligr
      cphasemeanconnect = black
      boxstyle          = pointsid
      phasemeansymbol   = dot
      readphase         = all
      phaselegend
      nolegend;
   label measure = 'Measurement'
         wafer   = 'Wafer Within Lot';
   id position;
run;
```

Output 39.7.3. Multi-Vari Chart Using BOXSTYLE=POINTSID



Chapter 40
CCHART Statement

Chapter Contents

OVERVIEW 1305

GETTING STARTED 1306
Creating c Charts from Defect Count Data 1306
Saving Control Limits 1308
Reading Preestablished Control Limits 1310
Creating c Charts from Nonconformities per Unit 1311
Saving Nonconformities per Unit 1313

SYNTAX 1315
Summary of Options 1317

DETAILS 1327
Constructing Charts for Numbers of Nonconformities (c Charts) 1327
Output Data Sets 1329
ODS Tables 1332
Input Data Sets 1332
Axis Labels 1336
Missing Values 1336

EXAMPLES 1337
Example 40.1. Applying Tests for Special Causes 1337
Example 40.2. Specifying a Known Expected Number of Nonconformities . 1340
Example 40.3. Creating c Charts for Varying Numbers of Units 1342

Chapter 40
CCHART Statement

Overview

The CCHART statement creates *c* charts for the numbers of nonconformities (defects) in subgroup samples.

You can use options in the CCHART statement to

- specify the number of inspection units per subgroup. Typically (but not necessarily), each subgroup consists of a single unit.
- compute control limits from the data based on a multiple of the standard error of the counts or as probability limits
- tabulate subgroup summary statistics and control limits
- save control limits in an output data set
- save subgroup summary statistics in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify a known (standard) value for the average number of nonconformities per inspection unit
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control the layout and appearance of the chart

Getting Started

This section introduces the CCHART statement with simple examples that illustrate commonly used options. Complete syntax for the CCHART statement is presented in the "Syntax" section on page 1315, and advanced examples are given in the "Examples" section on page 1337.

Creating c Charts from Defect Count Data

See SHWCCHR1 in the SAS/QC Sample Library

A *c* chart is used to monitor the number of paint defects on new trucks. Twenty trucks of the same model are inspected, and the number of paint defects per truck is recorded. The following statements create a SAS data set named TRUCKS, which contains the defect counts:

```
data trucks;
   input truckid $ defects @@;
   label truckid='Truck Identification Number'
         defects='Number of Paint Defects';
   datalines;
C1   5   C2   4   C3   4   C4   8   C5  7
C6  12   C7   3   C8  11   E4   8   E9  4
E7   9   E6  13   A3   5   A4   4   A7  9
Q1  15   Q2   8   Q3   9   Q9  10   Q4  8
;
run;
```

A partial listing of TRUCKS is shown in Figure 40.1.

Paint Defects on New Trucks

truckid	defects
C1	5
C2	4
C3	4
C4	8
C5	7
.	.
.	.
.	.

Figure 40.1. The Data Set TRUCKS

There is a single observation per truck. The variable TRUCKID identifies the subgroup sample and is referred to as the *subgroup-variable*. The variable DEFECTS contains the number of nonconformities in each subgroup sample and is referred to as the *process variable* (or *process* for short).

The following statements create the *c* chart shown in Figure 40.2:

```
title 'c Chart for Paint Defects on New Trucks';
proc shewhart data=trucks lineprinter;
   cchart defects*truckid='*';
run;
```

This example illustrates the basic form of the CCHART statement. After the keyword CCHART, you specify the *process* to analyze (in this case, DEFECTS) followed by an asterisk and the *subgroup-variable* (TRUCKID).

Since the LINEPRINTER option is specified in the PROC SHEWHART statement, line printer output is produced. The asterisk (*) specified in single quotes after the *subgroup-variable* specifies the *character* used to plot points. Note that this character must follow an equal sign.

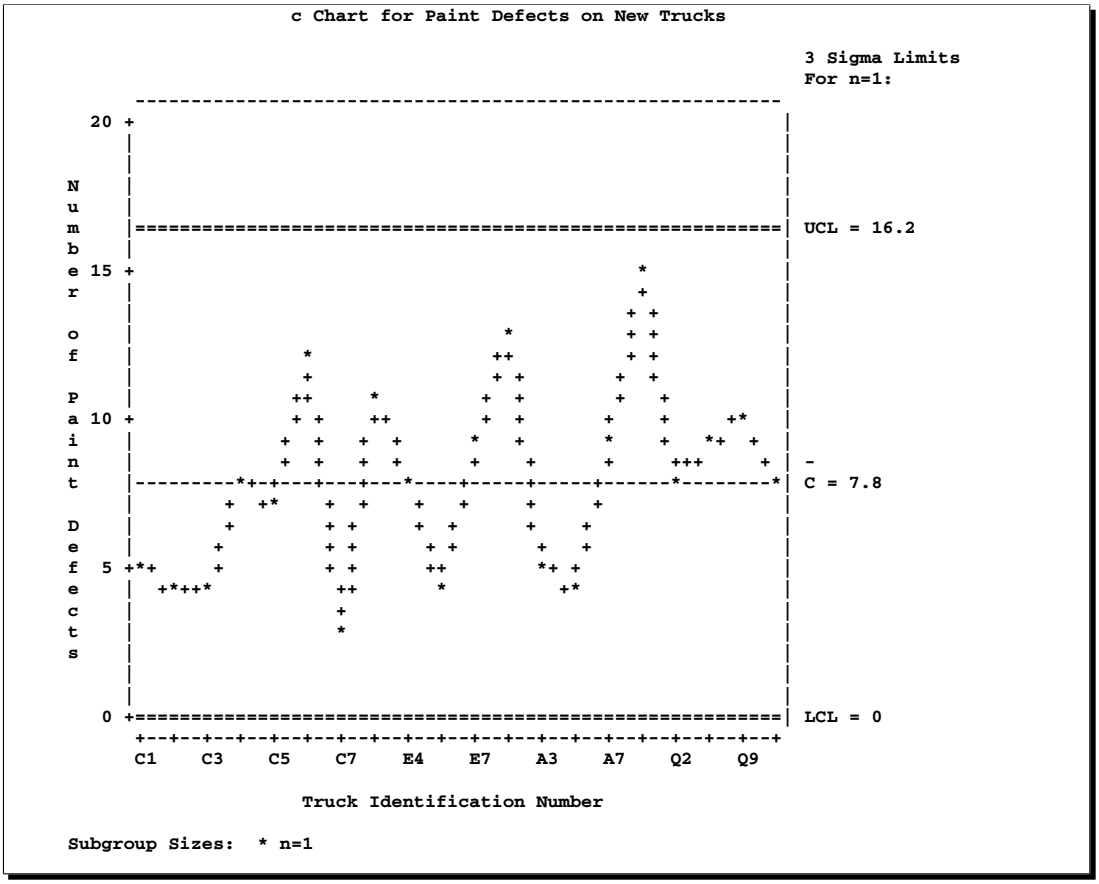


Figure 40.2. A c Chart of Paint Defects

Each point on the c chart represents the number of nonconformities for a particular subgroup. For instance, the value plotted for the first subgroup is 5 (since there are five paint defects on the first truck). By default, the control limits shown are 3σ limits estimated from the data; the formulas are given on page 1328. Since none of the points exceed the 3σ limits, the c chart indicates that the painting process is in statistical control.

See "Constructing Charts for Numbers of Nonconformities (c Charts)" on page 1327 for details concerning c charts. For more details on reading raw data, see "DATA= Data Set" on page 1332.

Saving Control Limits

See SHWCCHR1 in the SAS/QC Sample Library

You can save the control limits for a c chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1310) or subsequently modify the limits with a DATA step program.

The following statements read the data set TRUCKS introduced on page 1306 and saves the control limit information displayed in Figure 40.2 in a data set named DEFLIM:

```
proc shewhart data=trucks;
   cchart defects*truckid / outlimits=deflim
                            nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. Options such as OUTLIMITS= and NOCHART are specified after the slash (/) in the CCHART statement. A complete list of options is presented in the "Syntax" section on page 1315. The data set DEFLIM is listed in Figure 40.3.

Control Limits Data Set DEFLIM

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_U_	_LCLC_	_C_	_UCLC_
defects	truckid	ESTIMATE	1	.002902622	3	7.8	0	7.8	16.1785

Figure 40.3. The Data Set DEFLIM Containing Control Limit Information

The data set DEFLIM contains one observation with the limits for the *process* DEFECTS. The variables _LCLC_, and _UCLC_ contain the lower and upper control limits. The variable _C_ contains the central line, and the variable _U_ contains the average number of nonconformities per inspection unit. Since all the subgroups contain a single inspection unit, the values of _C_ and _U_ are the same. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the value of _U_ is an estimate or standard value. For more information, see "OUTLIMITS= Data Set" on page 1329.

Alternatively, you can use the OUTTABLE= option to create an output data set that saves both the control limits and the subgroup statistics, as illustrated by the following statements:

```
title 'Number of Nonconformities and Control Limit Information';
proc shewhart data=trucks;
   cchart defects*truckid / outtable=trucktab
                            nochart;
run;
```

The OUTTABLE= data set TRUCKTAB is listed in Figure 40.4.

Number of Nonconformities and Control Limit Information

VAR	truckid	_SIGMAS_	_LIMITN_	_SUBN_	_LCLC_	_SUBC_	_C_	_UCLC_	_EXLIM_
defects	C1	3	1	1	0	5	7.8	16.1785	
defects	C2	3	1	1	0	4	7.8	16.1785	
defects	C3	3	1	1	0	4	7.8	16.1785	
defects	C4	3	1	1	0	8	7.8	16.1785	
defects	C5	3	1	1	0	7	7.8	16.1785	
defects	C6	3	1	1	0	12	7.8	16.1785	
defects	C7	3	1	1	0	3	7.8	16.1785	
defects	C8	3	1	1	0	11	7.8	16.1785	
defects	E4	3	1	1	0	8	7.8	16.1785	
defects	E9	3	1	1	0	4	7.8	16.1785	
defects	E7	3	1	1	0	9	7.8	16.1785	
defects	E6	3	1	1	0	13	7.8	16.1785	
defects	A3	3	1	1	0	5	7.8	16.1785	
defects	A4	3	1	1	0	4	7.8	16.1785	
defects	A7	3	1	1	0	9	7.8	16.1785	
defects	Q1	3	1	1	0	15	7.8	16.1785	
defects	Q2	3	1	1	0	8	7.8	16.1785	
defects	Q3	3	1	1	0	9	7.8	16.1785	
defects	Q9	3	1	1	0	10	7.8	16.1785	
defects	Q4	3	1	1	0	8	7.8	16.1785	

Figure 40.4. The Data Set TRUCKTAB

This data set contains one observation for each subgroup sample. The variables _SUBC_ and _SUBN_ contain the number of nonconformities per subgroup and the number of inspection units per subgroup. The variables _LCLC_ and _UCLC_ contain the lower and upper control limits, and the variable _C_ contains the central line. The variables _VAR_ and TRUCKID contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1331.

An OUTTABLE= data set can be read later as a TABLE= data set in the SHEWHART procedure. For example, the following statements read TRUCKTAB and display a c chart (not shown here) identical to the chart in Figure 40.2:

```
title 'c Chart for Paint Defects in New Trucks';
proc shewhart table=trucktab;
   cchart defects*truckid='*';
label _SUBC_ = 'Number of Paint Defects';
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see Chapter 56, "Specialized Control Charts,"). For more information, see "TABLE= Data Set" on page 1334.

Reading Preestablished Control Limits

See SHWCCHR1 in the SAS/QC Sample Library

In the previous example, control limits were saved in a SAS data set named DEFLIM. This example shows how these limits can be applied to defect data for a second group of trucks, which are provided in the following data set:

```
data trucks2;
   input truckid $ defects @@;
   label truckid='Truck Identification Number'
         defects='Number of Paint Defects';
   datalines;
R1  7   R2   3   R3   3   R4  14   R5  7
R6  9   R7   3   R8  11   S4   6   S9  8
S7  3   S6  10   T3   9   T4   4   T7  7
X1  7   X2  18   X3   3   X9  11   X4  5
;
run;
```

The following statements plot the number of paint defects for the second group of trucks on a c chart using the control limits in DEFLIM. The chart is shown in Figure 40.5.

```
symbol v = dot;
title 'c Chart for Paint Defects in New Trucks';
proc shewhart data=trucks2 limits=deflim;
   cchart defects*truckid;
run;
```

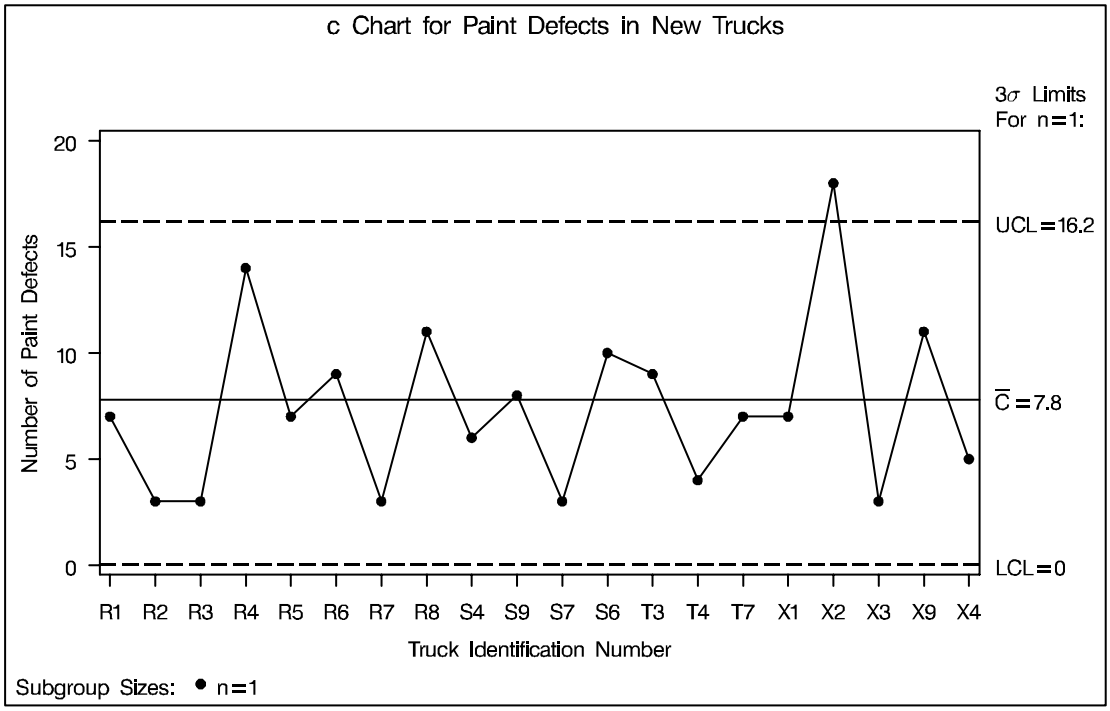


Figure 40.5. A c Chart for the Second Set of Trucks

Note that the number of defects on the truck with identification number X2 exceeds the upper control limit, indicating that the process is out-of-control. The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default,* this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name DEFECTS
- the value of _SUBGRP_ matches the *subgroup-variable* name TRUCKID

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1333 for details concerning the variables that you must provide.

Creating c Charts from Nonconformities per Unit

See SHWCCHR1 in the SAS/QC Sample Library

In the previous example, the input data set provided the number of nonconformities per subgroup sample. However, in some applications, as illustrated here, the data may be provided as the number of nonconformities *per inspection unit* for each subgroup.

A clothing manufacturer ships shirts in boxes of ten. Prior to shipment, each shirt is inspected for flaws. Since the manufacturer is interested in the average number of flaws per shirt, the number of flaws found in each box is divided by ten and then recorded. The following statements create a SAS data set named SHIRTS, which contains the average number of flaws per shirt for 25 boxes:

```
data shirts;
   input box avgdefu @@;
   avgdefn=10;
   datalines;
 1  0.4   2  0.7   3  0.5   4  1.0   5  0.3
 6  0.2   7  0.0   8  0.4   9  0.4  10  0.6
11  0.2  12  0.7  13  0.3  14  0.1  15  0.3
16  0.6  17  0.6  18  0.3  19  0.7  20  0.3
21  0.0  22  0.1  23  0.5  24  0.6  25  0.4
;
run;
```

A partial listing of SHIRTS is shown in Figure 40.6.

The data set SHIRTS contains three variables: the box number (BOX), the average number of flaws per shirt (AVGDEFU), and the number of shirts per box (AVGDEFN). Here, a *subgroup* is a box of shirts, and an *inspection unit* is an individual shirt. Note that each subgroup consists of ten inspection units.

To create a c chart plotting the total number of flaws per box (instead of per shirt), you can specify SHIRTS as a HISTORY= data set.

*In Release 6.09 and in earlier releases, it is also necessary to specify the READLIMITS option to read control limits from a LIMITS= data set.

Average Number of Shirt Flaws

box	avgdefu	avgdefn
1	0.4	10
2	0.7	10
3	0.5	10
4	1.0	10
.	.	.
.	.	.
.	.	.

Figure 40.6. The Data Set SHIRTS

```
symbol h = .8;
title 'Total Flaws per Box of Shirts';
proc shewhart history=shirts;
   cchart avgdef*box ;
run;
```

Note that AVGDEF is *not* the name of a SAS variable in the data set but is instead the common prefix for the SAS variable names AVGDEFU and AVGDEFN. The suffix characters *U* and *N* indicate *number of nonconformities per unit* and *sample size*, respectively. This naming convention enables you to specify two variables in the HISTORY= data set with a single name referred to as the *process*. The name BOX specified after the asterisk is the name of the *subgroup-variable*. The c chart is shown in Figure 40.7.

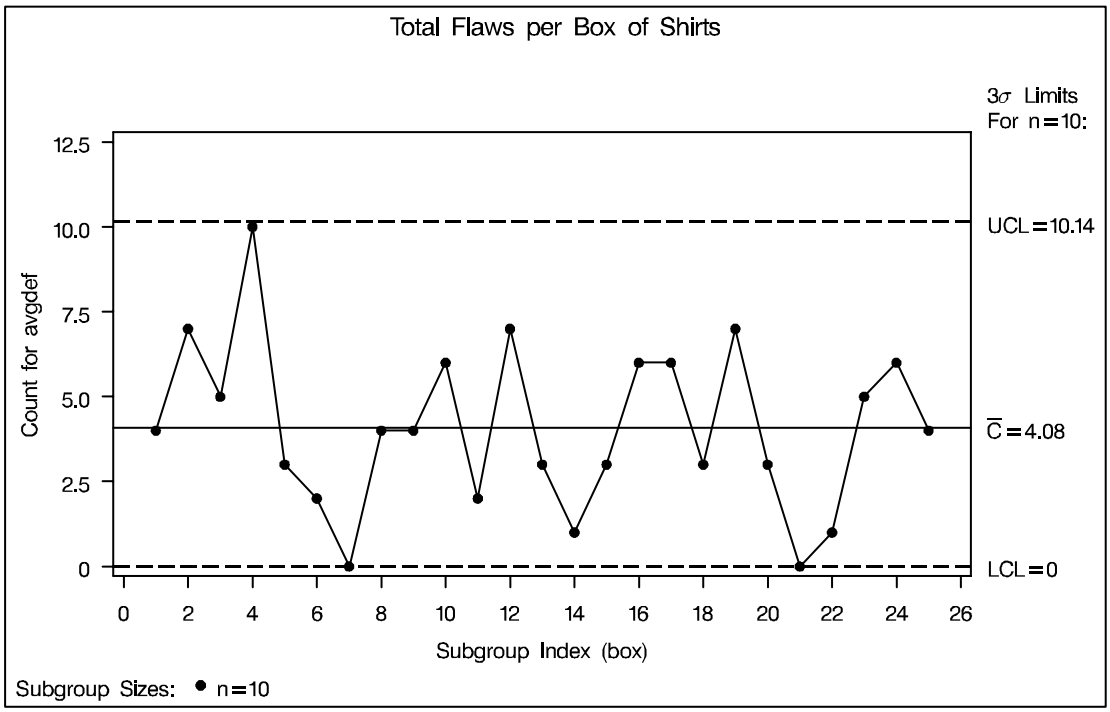


Figure 40.7. A c Chart for Boxes of Shirts

In general, a HISTORY= input data set used with the CCHART statement must contain the following variables:

- subgroup variable
- subgroup number of nonconformities per unit variable
- subgroup sample size variable

Furthermore, the names of the nonconformities per unit and sample size variables must begin with the *process* name specified in the CCHART statement and end with the special suffix characters *U* and *N*, respectively. If the names do not follow this convention, you can use the RENAME option to rename the variables for the duration of the SHEWHART procedure step. Suppose that, instead of the variables AVGDEFU and AVGDEFN, the data set SHIRTS contained the variables SHIRTDEF and SIZES. The following statements would temporarily rename SHIRTDEF and SIZES to AVGDEFU and AVGDEFN:

```
proc shewhart
   history=shirts (rename=(shirtdef = avgdefu
                           sizes    = avgdefn ));
   cchart avgdef*box;
run;
```

For more information, see "HISTORY= Data Set" on page 1333.

Saving Nonconformities per Unit

A department store receives boxes of shirts containing 10, 25, or 50 shirts. Each box is inspected, and the total number of defects per box is recorded. The following statements create a SAS data set named SHIRTS2, which contains the total defects per box for 20 boxes:

See SHWCCHR1 in the SAS/QC Sample Library

```
data shirts2;
   input box flaws nshirts @@;
   datalines;
 1  3  10    2  8  10    3 15  25    4 20  25
 5  9  25    6  1  10    7  1  10    8 21  50
 9  3  10   10  7  10   11  1  10   12 21  25
13  9  25   14  3  25   15 12  50   16 18  50
17  7  10   18  4  10   19  8  10   20  4  10
;
run;
```

A partial listing of SHIRTS2 is shown in Figure 40.8.

Number of Shirt Flaws per Box

box	flaws	nshirts
1	3	10
2	8	10
3	15	25
4	20	25
5	9	25
.	.	.
.	.	.
.	.	.

Figure 40.8. The Data Set SHIRTS2

The variable BOX contains the box number, the variable FLAWS contains the number of flaws in each box, and the variable NSHIRTS contains the number of shirts in each box. To evaluate the quality of the shirts, you should report the average number of defects per shirt. The following statements create a data set containing the number of flaws per shirt and the number of shirts per box:

```
proc shewhart data=shirts2;
   cchart flaws*box / subgroupn  = nshirts
                      outhistory = shirthist
                      nochart ;
run;
```

The SUBGROUPN= option names the variable in the DATA= data set whose values specify the number of inspection units per subgroup. The OUTHISTORY= option names an output data set containing the number of nonconformities per inspection unit and the number of inspection units per subgroup. A partial listing of SHIRTHIST is shown in Figure 40.9.

Average Defects Per Shirt

box	flaws U	flaws N
1	0.30	10
2	0.80	10
3	0.60	25
4	0.80	25
5	0.36	25
.	.	.
.	.	.
.	.	.

Figure 40.9. The Data Set SHIRTHIST

There are three variables in the data set SHIRTHIST.

- BOX contains the subgroup index.
- FLAWSU contains the numbers of nonconformities per inspection unit.
- FLAWSN contains the subgroup sample sizes.

Note that the variables containing the numbers of nonconformities per inspection unit and subgroup sample sizes are named by adding the suffix characters *U* and *N* to the *process* DEFECTS specified in the CCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 1330.

Syntax

The basic syntax for the CCHART statement is as follows:

CCHART *process***subgroup-variable* **;**

The general form of this syntax is as follows:

CCHART (*processes***)****subgroup-variable* <**(***block-variables* **)** >
< =*symbol-variable* | =*'character'* > < **/** *options* >**;**

You can use any number of CCHART statements in the SHEWHART procedure. The components of the CCHART statement are described as follows.

process
processes
identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If numbers of nonconformities per subgroup are read from a DATA= data set, *process* must be the name of the variable containing the numbers of nonconformities.

  For an example, see "Creating c Charts from Defect Count Data" on page 1306.

- If numbers of nonconformities per unit and numbers of inspection units per subgroup are read from a HISTORY= data set, *process* must be the common prefix of the appropriate variables in the HISTORY= data set.

  For an example, see "Creating c Charts from Nonconformities per Unit" on page 1311.

- If numbers of nonconformities per subgroup, numbers of inspection units per subgroup, and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set.

  For an example, see "Saving Control Limits" on page 1308.

A *process* is required. If you specify more than one process, enclose the list in parentheses. For example, the following statements request distinct c charts for DEFECTS and FLAWS:

```
proc shewhart data=info;
   cchart (defects flaws)*sample;
run;
```

subgroup-variable
is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding CCHART statement, SAMPLE is the subgroup variable. For details, see "Subgroup Variables" on page 1771.

block-variables
are optional variables that group the data into blocks of consecutive subgroups. These blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 1932 for an example.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the number of nonconformities.

- If you produce a chart on a line printer, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce a chart on a graphics device, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOLn statements. See "Displaying Stratification in Levels of a Classification Variable" on page 1931 for an example.

character
specifies a plotting character for charts produced on line printers. For example, the following statements create a c chart using an asterisk (*) to plot the points:

```
proc shewhart data=info;
   cchart defects*sample='*';
run;
```

options
enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The "Summary of Options" section, which follows, lists all options by function. Chapter 53, "Dictionary of Options," describes each option in detail.

Summary of Options

The following tables list the CCHART statement options by function. For complete descriptions, see Chapter 53, "Dictionary of Options."

Table 40.1. Tabulation Options

TABLE	creates a basic table of subgroup sample sizes, subgroup numbers of nonconformities, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with a column indicating which tests for special causes are positive

Note that specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.

Table 40.2. Reference Line Options

CHREF=*color*	specifies color for lines requested by the HREF= option
CVREF=*color*	specifies color for lines requested by the VREF= option
HREF=*values*|*SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis
HREFCHAR='*character*'	specifies line character for HREF= lines
HREFDATA=*SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis
HREFLABELS=*'label1'...'labeln'*	specifies labels for HREF= lines
HREFLABPOS=*n*	specifies position of HREFLABELS= labels
LHREF=*linetype*	specifies line type for HREF= lines
LVREF=*linetype*	specifies line type for VREF= lines
NOBYREF	specifies that reference line information in a data set is to be applied uniformly to charts created for all BY groups
VREF=*values*|*SAS-data-set*	specifies position of reference lines perpendicular to vertical axis
VREFCHAR='*character*'	specifies line character for VREF= lines
VREFLABELS=*'label1'...'labeln'*	specifies labels for VREF= lines
VREFLABPOS=*n*	specifies position of VREFLABELS= labels

Table 40.3. Options for Specifying Tests for Special Causes

NO3SIGMACHECK	allows tests to be applied with control limits other than 3σ limits
TESTS=*value-list*| *customized-pattern-list*	specifies tests for special causes
TEST2RUN=*n*	specifies length of pattern for Test 2
TEST3RUN=*n*	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL='*label*'| (*variable*)|*keyword*	provides labels for points where test is positive
TESTLABEL*n*='*label*'	specifies label for n^{th} test for special causes
TESTNMETHOD= STANDARDIZE	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
TESTRESET=*variable*	allows tests for special causes to be reset
ZONELABELS	adds labels A, B, and C to zone lines
ZONES	adds lines delineating zones A, B, and C
ZONEVALPOS=*n*	specifies position of ZONEVALUES labels
ZONEVALUES	labels zone lines with their values

Table 40.4. Graphical Options for Displaying Tests for Special Causes

CTESTLABBOX=*color*	specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=*color*| *test-color-list*	specifies color for labels indicating points where test is positive
CTESTSYMBOL=*color*	specifies color for symbol used to plot points where test is positive
CZONES=*color*	specifies color for lines and labels delineating zones A, B, and C
LTESTS=*linetype*	specifies type of line connecting points where test is positive
LZONES=*linetype*	specifies line type for lines delineating zones A, B, and C
TESTFONT=*font*	specifies software font for labels at points where test is positive
TESTHEIGHT=*value*	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=*symbol*	specifies plot symbol for points where test is positive
TESTSYMBOLHT=*value*	specifies symbol height for points where test is positive
WTESTS=*n*	specifies width of line connecting points where test is positive

Table 40.5. Line Printer Options for Displaying Tests for Special Causes

TESTCHAR='*character*'	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
ZONECHAR='*character*'	specifies character for lines that delineate zones for tests for special causes

Table 40.6. Options for Plotting and Labeling Points

ALLLABEL=VALUE|(*variable*)	labels every point
CLABEL=*color*	specifies color for labels
CCONNECT=*color*	specifies color for line segments that connect points on chart
CFRAMELAB=*color*	specifies fill color for frame around labeled points
CNEEDLES=*color*	specifies color for needles that connect points to central line
CONNECTCHAR='*character*'	specifies character used to form line segments that connect points on chart
COUT=*color*	specifies color for portions of line segments that connect points outside control limits
COUTFILL=*color*	specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=*angle*	specifies angle at which labels are drawn
LABELFONT=*font*	specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=*value*	specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on chart
OUTLABEL=VALUE|(*variable*)	labels points outside control limits
SYMBOLCHARS='*characters*'	specifies characters indicating *symbol-variable*
SYMBOLLEGEND=NONE|*name*	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=*keyword*	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=*n*	specifies width of needles

Table 40.7. Clipping Options

CCLIP=*color*	specifies color for plot symbol for clipped points
CLIPCHAR='*character*'	specifies plot character for clipped points
CLIPFACTOR=*value*	determines extent to which extreme points are clipped
CLIPLEGEND='*string*'	specifies text for clipping legend
CLIPLEGPOS=*keyword*	specifies position of clipping legend
CLIPSUBCHAR='*character*'	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=*symbol*	specifies plot symbol for clipped points
CLIPSYMBOLHT=*value*	specifies symbol marker height for clipped points

Table 40.8. Standard Value Options

TYPE=*keyword*	identifies whether parameters are estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
U0=*value*	specifies known average number of nonconformities per unit

Table 40.9. Axis and Axis Label Options

CAXIS=*color*	specifies color for axis lines and tick marks
CFRAME=*color*|(*color-list*)	specifies fill colors for frame for plot area
CTEXT=*color*	specifies color for tick mark values and axis labels
HAXIS=*values*|AXISn	specifies major tick mark values for horizontal axis
HEIGHT=*value*	specifies height of axis label and axis legend text
HMINOR=*n*	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=*value*	specifies length of offset at both ends of horizontal axis
INTSTART=*value*	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	suppresses vertical axis truncation at zero applied by default
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=*n*	specifies thinning factor for tick mark labels on horizontal axis
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=*values*|AXISn	specifies major tick mark values for vertical axis
VFORMAT=*format*	specifies format for vertical axis tick mark labels
VMINOR=*n*	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=*value*	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis
WAXIS=*n*	specifies width of axis lines

Table 40.10. Input Data Set Options

MISSBREAK	specifies that observations with missing values are not to be processed
SUBGROUPN=*n*|*variable*	specifies subgroup sample sizes as constant number n or as values of *variable* in a DATA= data set

Table 40.11. Output Data Set Options

OUTHISTORY= *SAS-data-set*	creates output data set containing subgroup numbers of nonconformities per unit and subgroup sample sizes
OUTINDEX='*string*'	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS= *SAS-data-set*	creates output data set containing control limits
OUTTABLE= *SAS-data-set*	creates output data set containing subgroup numbers of nonconformities, subgroup sample sizes, and control limits

Table 40.12. Options for Specifying Control Limits

ALPHA=*value*	requests probability limits for control charts
LIMITN=*n*|VARYING	specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	computes control limits for each *process* from the data rather than from a LIMITS= data set (Release 6.10 and later releases)
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEXES=ALL| '*label1*'...'*labeln*'	reads multiple sets of control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (Release 6.09 and earlier releases)
SIGMAS=*k*	specifies width of control limits in terms of multiple *k* of standard error of plotted statistic

Table 40.13. Options for Displaying Control Limits

CINFILL=*color*	specifies color for area inside control limits
CLIMITS=*color*	specifies color of control limits, central line, and related labels
CSYMBOL='*string*'| *keyword*	specifies label for central line
LCLLABEL='*label*'	specifies label for lower control limit
LIMLABSUBCHAR= '*character*'	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=*linetype*	specifies line type for control limits
NDECIMAL=*n*	specifies number of digits to right of decimal place in default labels for control limits and central line
NOCTL	suppresses display of central line
NOLCL	suppresses display of lower control limit
NOLIMITLABEL	suppresses labels for control limits and central line
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOLIMIT0	suppresses display of lower control limit if it is 0
NOUCL	suppresses display of upper control limit
UCLLABEL='*string*'	specifies label for upper control limit
WLIMITS=*n*	specifies width for control limits and central line

Table 40.14. Phase Options

CPHASELEG=*color*	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE='*string*'	specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=*value*| *keyword*	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES= ALL| '*label1*'...'*labeln*'	specifies *phases* to be read from an input data set

Table 40.15. Options for Interactive Control Charts

HTML=(*variable*)	specifies a variable whose values are URLs to be associated with subgroups
HTML_LEGEND=(*variable*)	specifies a variable whose values are URLs to be associated with symbols in the symbol legend
WEBOUT=*SAS-data-set*	creates an OUTTABLE= data set with additional graphics coordinate data

Table 40.16. Block Variable Legend Options

BLOCKLABELPOS=*keyword*	specifies position of label for *block-variable* legend
BLOCKLABTYPE=*value*|*keyword*	specifies text size of *block-variable* legend
BLOCKPOS=*n*	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=*color-list*	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=*variable*|(*variables*)	specifies one or more variables whose values are colors for filling background of *block-variable* legend

Table 40.17. Plot Layout Options

ALLN	plots numbers of nonconformities for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts for a process variable only when exceptions occur
INTERVAL=*keyword*	specifies natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=*n*	specifies maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=*n*	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position on next panel
TOTPANELS=*n*	specifies number of pages or screens to be used to display chart
ZEROSTD	displays c chart regardless of whether $\hat{\sigma} = 0$

Table 40.18. Graphical Enhancement Options

ANNOTATE= *SAS-data-set*	specifies annotate data set that adds features to chart
DESCRIPTION='*string*'	specifies string that appears in the description field of the PROC GREPLAY master menu
FONT=*font*	specifies software font for labels and legends on charts
NAME='*string*'	specifies name that appears in the name field of the PROC GREPLAY master menu
PAGENUM='*string*'	specifies the form of the label used in pagination
PAGENUMPOS= *keyword*	specifies the position of the page number requested with the PAGENUM= option

Table 40.19. Grid Options

CGRID=*color*	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=*linetype*	specifies line type for grid requested with the ENDGRID option
LGRID=*linetype*	specifies line type for grid requested with the GRID option
WGRID=*n*	specifies width of grid lines

Table 40.20. Star Options

CSTARCIRCLES=*color*	specifies color for circles specified by the STARCIRCLES= option
CSTARFILL=*color*| *(variable)*	specifies color for filling stars
CSTAROUT=*color*	specifies outline color for stars exceeding inner or outer circles
CSTARS=*color*| *(variable)*	specifies color for outlines of stars
LSTARCIRCLES= *linetypes*	specifies line types for the circles requested with the STARCIRCLES= option
LSTARS=*linetype*| *(variable)*	specifies line types for outlines of stars requested with the STARVERTICES= option
STARBDRADIUS=*value*	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=*value-list*	specifies reference circles for stars
STARINRADIUS=*value*	specifies inner radius of stars
STARLABEL=*keyword*	specifies vertices to be labeled
STARLEGEND=*keyword*	specifies style of legend for star vertices
STARLEGENDLAB= *'label'*	specifies label for STARLEGEND= legend
STAROUTRADIUS=*value*	specifies outer radius of stars
STARSPEC=*value*| *SAS-data-set*	specifies method used to standardize vertex variables
STARSTART=*value*	specifies angle for first vertex
STARTYPE=*keyword*	specifies graphical style of star
STARVERTICES= *variable*|*(variables)*	superimposes star at each point on chart
WSTARCIRCLES=*n*	specifies width of circles requested by the STARCIRCLES= option
WSTARS=*n*	specifies width of stars requested by the STARVERTICES= option

Table 40.21. Overlay Options

CCOVERLAY=*color-list*	specifies colors for overlay line segments
COVERLAY=*color-list*	specifies colors for overlay plots
COVERLAYCLIP=*color*	specifies color for clipped points on overlays
LOVERLAY=*linetypes*	specifies line types for overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=*variable-list*	specifies variables to overlay on control chart
OVERLAYCLIPSYM=*symbol*	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=*value*	specifies symbol height for clipped points on overlays
OVERLAYHTML=*variable-list*	specifies URLS to associate with overlay points
OVERLAYID=*variable-list*	specifies labels for overlay points
OVERLAYLEGLAB='*label*'	specifies label for overlay legend
OVERLAYSYM=*symbol-list*	specifies symbols for overlay plots
OVERLAYSYMHT=*value-list*	specifies symbol heights for overlay plots
WOVERLAY=*value-list*	specifies widths of overlay line segments

Details

Constructing Charts for Numbers of Nonconformities (c Charts)

The following notation is used in this section:

u	expected number of nonconformities per unit produced by the process
u_i	number of nonconformities per unit in the i^{th} subgroup
c_i	total number of nonconformities in the i^{th} subgroup
n_i	number of inspection units in the i^{th} subgroup. Typically, $n_i = 1$ and $u_i = c_i$ for c charts. In general, $u_i = c_i/n_i$.
$\bar{u}$	average number of nonconformities per unit taken across subgroups. The quantity $\bar{u}$ is computed as a weighted average: $$\bar{u} = \frac{n_1 u_1 + \cdots + n_N u_N}{n_1 + \cdots + n_N} = \frac{c_1 + \cdots + c_N}{n_1 + \cdots + n_N}$$
N	number of subgroups
χ^2_ν	has a central χ^2 distribution with ν degrees of freedom

Plotted Points

Each point on a c chart represents the total number of nonconformities (c_i) in a subgroup. For example, Figure 40.10 displays three sections of pipeline that are inspected for defective welds (indicated by an *X*). Each section represents a *subgroup* composed of a number of *inspection units*, which are 1000-foot-long sections. The number of units in the i^{th} subgroup is denoted by n_i, which is the subgroup sample size. The value of n_i can be fractional; Figure 40.10 shows $n_3 = 2.5$ units in the third subgroup.

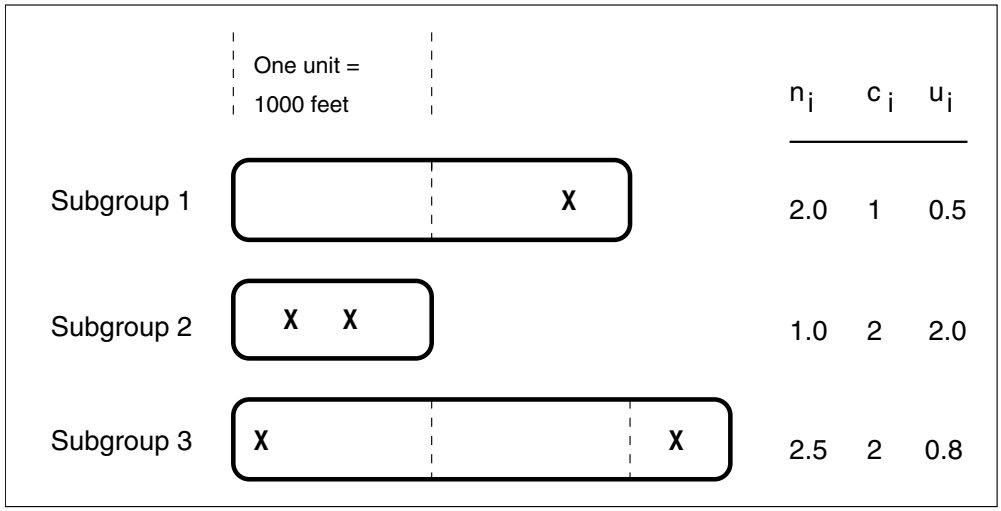


Figure 40.10. Terminology for c Charts and u Charts

The *number of nonconformities* in the i^{th} subgroup is denoted by c_i. The *number of nonconformities per unit* in the i^{th} subgroup is denoted by $u_i = c_i/n_i$. In Figure 40.10, the number of welds per inspection unit in the third subgroup is $u_3 = 2/2.5 = 0.8$.

A u chart created with the UCHART statement plots the quantity u_i for the i^{th} subgroup (see Chapter 48). An advantage of a u chart is that the value of the central line at the i^{th} subgroup does not depend on n_i. This is not the case for a c chart, and consequently, a u chart is often preferred when the number of units n_i is not constant across subgroups.

Central Line

On a c chart, the central line indicates an estimate for $n_i u$, which is computed as $n_i \bar{u}$. If you specify a known value (u_0) for u, the central line indicates the value of $n_i u_0$.

Note that the central line varies with subgroup sample size n_i. When $n_i = 1$ for all subgroups, the central line has the constant value $\bar{c} = (c_1 + \cdots + c_N)/N$.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of c_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that c_i exceeds the limits

The lower and upper control limits, LCLC and UCLC respectively, are given by

$$\begin{aligned} \text{LCLC} &= \max\left(n_i\bar{u} - k\sqrt{n_i\bar{u}}\,, 0\right) \\ \text{UCLC} &= n_i\bar{u} + k\sqrt{n_i\bar{u}} \end{aligned}$$

The upper and lower control limits vary with the number of inspection units per subgroup n_i. If $n_i = 1$ for all subgroups, the control limits have constant values.

$$\begin{aligned} \text{LCLC} &= \max\left(\bar{c} - k\sqrt{\bar{c}}\,, 0\right) \\ \text{UCLC} &= \bar{c} + k\sqrt{\bar{c}} \end{aligned}$$

An upper probability limit UCLC for c_i can be determined using the fact that

$$\begin{aligned} P\{c_i > \text{UCLC}\} &= 1 - P\{c_i \le \text{UCLC}\} \\ &= 1 - P\{\chi^2_{2(\text{UCLC}+1)} \ge 2n_i\bar{u}\} \end{aligned}$$

The upper probability limit UCLC is then calculated by setting

$$1 - P\{\chi^2_{2(\text{UCLC}+1)} \ge 2n_i\bar{u}\} = \alpha/2$$

and solving for UCLC.

A similar approach is used to calculate the lower probability limit LCLC, using the fact that

$$P\{c_i < \text{LCLC}\} = P\{\chi^2_{2(\text{LCLC}+1)} > 2n_i\bar{u}\}$$

The lower probability limit LCLC is then calculated by setting

$$P\{\chi^2_{2(\text{LCLC}+1)} > 2n_i\bar{u}\} = \alpha/2$$

and solving for LCLC. This assumes that the process is in statistical control and that c_i has a Poisson distribution. For more information, refer to Johnson, Kotz, and Kemp (1992). Note that the probability limits vary with the number of inspection units per subgroup (n_i) and are asymmetric about the central line.

If a standard value u_0 is available for u, replace $\bar{u}$ with u_0 in the formulas for the control limits. You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify u_0 with the U0= option or with the variable _U_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables can be saved:

Table 40.22. OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
_C_	value of central line on c chart ($n_i\bar{u}$ or n_iu_0)
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLC	lower control limit for number of nonconformities
LIMITN	sample size associated with the control limits
SIGMAS	multiple (k) of standard error of c_i
SUBGRP	*subgroup-variable* specified in the CCHART statement
TYPE	type (estimate or standard value) of _U_
_U_	average number of nonconformities per unit ($\bar{u}$ or u_0)
UCLC	upper control limit for number of nonconformities
VAR	*process* specified in the CCHART statement

Notes:

1. If the control limits vary with subgroup sample size, the special missing value *V* is assigned to the variables _C_, _LCLC_, _UCLC_, and _LIMITN_.
2. If the limits are defined in terms of a multiple k of the standard error of c_i, the value of _ALPHA_ is computed as $P\{c_i < \text{\_LCLC\_}\} + P\{c_i > \text{\_UCLC\_}\}$. If control limits vary with subgroup sample size and are determined in terms of k, _ALPHA_ is assigned the special missing value *V*.
3. If the limits are probability limits, the value of _SIGMAS_ is computed as $(\text{\_UCLC\_} - \text{\_C\_})/\sqrt{\text{\_C\_}}$. If probability limits vary with subgroup sample size, _SIGMAS_ is assigned the special missing value *V*.
4. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the CCHART statement. For an example, see "Saving Control Limits" on page 1308.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup statistics. The following variables are saved:

- the *subgroup-variable*
- a subgroup sample size variable named by *process* suffixed with *N*
- a subgroup number of nonconformities per unit variable named by *process* suffixed with *U*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the CCHART statement. For example, consider the following statements:

```
proc shewhart data=fabric;
   cchart (flaws ndefects)*lot / outhistory=summary;
run;
```

The data set SUMMARY contains variables named LOT, FLAWSU, FLAWSN, NDEFCTSU, and NDEFCTSN. Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example that creates an OUTHISTORY= data set, see "Saving Nonconformities per Unit" on page 1313. Note that an OUTHISTORY= data set created with the CCHART statement can be used as a HISTORY= data set by either the CCHART statement or the UCHART statement.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables are saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on c chart
LCLC	lower control limit for number of nonconformities
LIMITN	nominal sample size associated with the control limits
SIGMAS	multiple (k) of the standard error associated with control limits
subgroup	values of the subgroup variable
SUBC	subgroup number of nonconformities
SUBN	subgroup sample size
TESTS	tests for special causes signaled on c chart
UCLC	upper control limit for number of nonconformities
VAR	*process* specified in the CCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The k^{th} character of a value of _TESTS_ is k if Test k is positive at that subgroup. For example, if you request the first four tests (the ones appropriate for c charts) and Tests 2 and 4 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, a 4 for the fourth character, and blanks for the other six characters.
3. The variables _EXLIM_ and _TESTS_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1308.

ODS Tables

The following table summarizes the ODS tables that you can request with the CCHART statement.

Table 40.23. ODS Tables Produced with the CCHART Statement

Table Name	Description	Options
CCHART	*c* chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

Input Data Sets

DATA= Data Set

You can read the number of nonconformities in subgroup samples from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the CCHART statement must be a SAS variable in the data set. This variable provides the number of nonconformities in subgroup samples indexed by the *subgroup-variable*. Typically (but not necessarily), the subgroup consists of a single inspection unit. The *subgroup-variable*, specified in the CCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. The data set must contain one observation per subgroup. Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 1936).

For an example of a DATA= data set, see "Creating c Charts from Defect Count Data" on page 1306.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set CONLIMS:*

```
proc shewhart data=info limits=conlims;
   cchart defects*lot;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLC_, _C_, and _UCLC_, which specify the control limits
- the variable _U_, which is used to calculate the control limits (see page 1328)

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are `ESTIMATE` and `STANDARD`.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1310.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This allows you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART procedure or to create your own HISTORY= data set. A HISTORY= data set used with the CCHART statement must contain the following variables:

- *subgroup-variable*
- subgroup number of nonconformities per unit variable for each *process*
- subgroup sample size variable (number of units per subgroup) for each *process*

*In Release 6.09 and in earlier releases, it is necessary to specify the READLIMITS option.

The names of the subgroup number of nonconformities per unit and subgroup sample size variables must be the *process* name concatenated with the special suffix characters U and N, respectively. For example, consider the following statements:

```
proc shewhart history=summary;
   cchart (flaws ndefects)*lot;
run;
```

The data set SUMMARY must include the variables LOT, FLAWSU, FLAWSN, NDEFCTSU, and NDEFCTSN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character. Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (see "Displaying Stratification in Phases" on page 1936 for an example).

For an example of a HISTORY= data set, see "Creating c Charts from Nonconformities per Unit" on page 1311.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure or to create your own TABLE= data set. Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in Chapter 56, "Specialized Control Charts."

The following table lists the variables required in a TABLE= data set used with the CCHART statement:

Table 40.24. Variables Required in a TABLE= Data Set

Variable	Description
_C_	average number of nonconformities
LCLC	lower control limit for nonconformities
LIMITN	nominal sample size associated with the control limits
subgroup-variable	values of the *subgroup-variable*
SUBC	subgroup number of nonconformities
SUBN	subgroup sample size
UCLC	upper control limit for nonconformities

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1308.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical	DATA=	*process*
Vertical	HISTORY=	subgroup defect counts variable
Vertical	TABLE=	_SUBC_

For an example, see "Labeling Axes" on page 1966.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples

This section provides advanced examples of the CCHART statement.

Example 40.1. Applying Tests for Special Causes

See SHWCEX1 in the SAS/QC Sample Library

This example illustrates how you can apply tests for special causes to make c charts more sensitive to special causes of variation. Twenty trucks of the same model are inspected, and the number of paint defects per truck is recorded. The following statements create a SAS data set named TRUCKS3:

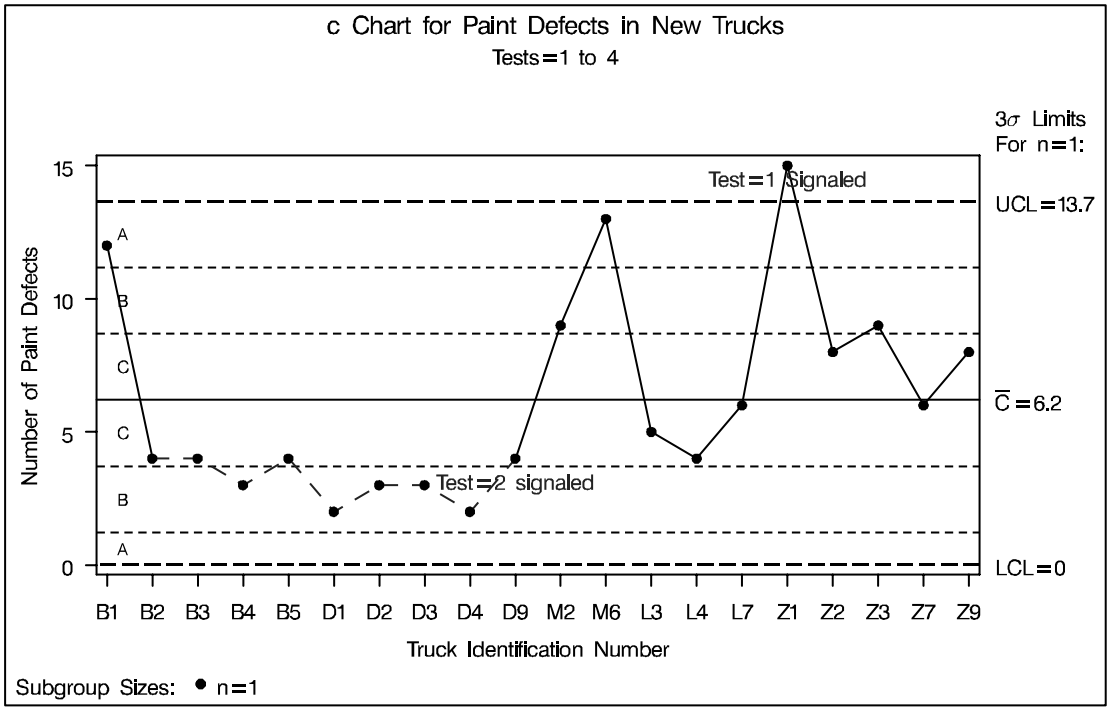
```
data trucks3;
   input truckid $ defects @@;
   label truckid='Truck Identification Number'
         defects='Number of Paint Defects';
   datalines;
B1  12    B2  4    B3  4    B4   3
B5   4    D1  2    D2  3    D3   3
D4   2    D9  4    M2  9    M6  13
L3   5    L4  4    L7  6    Z1  15
Z2   8    Z3  9    Z7  6    Z9   8
;
run;
```

The following statements create a c chart and tabulate the information on the chart. The chart and table are shown in Output 40.1.1 and Output 40.1.2.

```
symbol h = .8;
title1 'c Chart for Paint Defects in New Trucks';
title2 'Tests=1 to 4';
proc shewhart data=trucks3;
   cchart defects*truckid / tests      = 1 to 4
                            testlabel1 = 'Test=1 Signaled'
                            testlabel2 = 'Test=2 signaled'
                            testfont   = swiss
                            zonelabels
                            ltests     = 20
                            tabletests
                            tablelegend;
run;
```

The TESTS= option requests Tests 1, 2, 3, and 4, which are described in Chapter 55, "Tests for Special Causes." Only Tests 1, 2, 3, and 4 are recommended for c charts. The TESTLABEL1= and TESTLABEL2= options specify the labels for points where Tests 1 and 2 are positive. The TESTFONT= option specifies the font for the labels indicating points at which the tests are positive.

Output 40.1.1. Tests for Special Causes Displayed on c Chart



Output 40.1.2. Tabular Form of c Chart

c Chart for Paint Defects in New Trucks
Tests=1 to 4

c Chart Summary for defects

truckid	Subgroup Sample Size	-3 Sigma Limits with n=1 for Count- Lower Limit	Subgroup Count	Upper Limit	Special Tests Signaled
B1	1.00000	0	12.000000	13.669940	
B2	1.00000	0	4.000000	13.669940	
B3	1.00000	0	4.000000	13.669940	
B4	1.00000	0	3.000000	13.669940	
B5	1.00000	0	4.000000	13.669940	
D1	1.00000	0	2.000000	13.669940	
D2	1.00000	0	3.000000	13.669940	
D3	1.00000	0	3.000000	13.669940	
D4	1.00000	0	2.000000	13.669940	
D9	1.00000	0	4.000000	13.669940	2
M2	1.00000	0	9.000000	13.669940	
M6	1.00000	0	13.000000	13.669940	
L3	1.00000	0	5.000000	13.669940	
L4	1.00000	0	4.000000	13.669940	
L7	1.00000	0	6.000000	13.669940	
Z1	1.00000	0	15.000000	13.669940	1
Z2	1.00000	0	8.000000	13.669940	
Z3	1.00000	0	9.000000	13.669940	
Z7	1.00000	0	6.000000	13.669940	
Z9	1.00000	0	8.000000	13.669940	

Test Descriptions

Test 1 One point beyond Zone A (outside control limits)
Test 2 Nine points in a row on one side of center line

The ZONELABELS option requests zone lines and displays zone labels on the chart. The zones are used to define the tests. The LTESTS= option specifies the line type used to connect the points in a pattern for a test that is signaled. The TABLETESTS option requests a table of counts of nonconformities, subgroup sample sizes, and control limits, together with a column indicating the subgroups at which the tests are positive. The TABLELEGEND option adds a legend describing the tests that are positive.

Output 40.1.1 and Output 40.1.2 indicate that Test 1 is positive at Truck Z1 and Test 2 is positive at Truck D9.

Example 40.2. Specifying a Known Expected Number of Nonconformities

See SHWCEX2 in the SAS/QC Sample Library

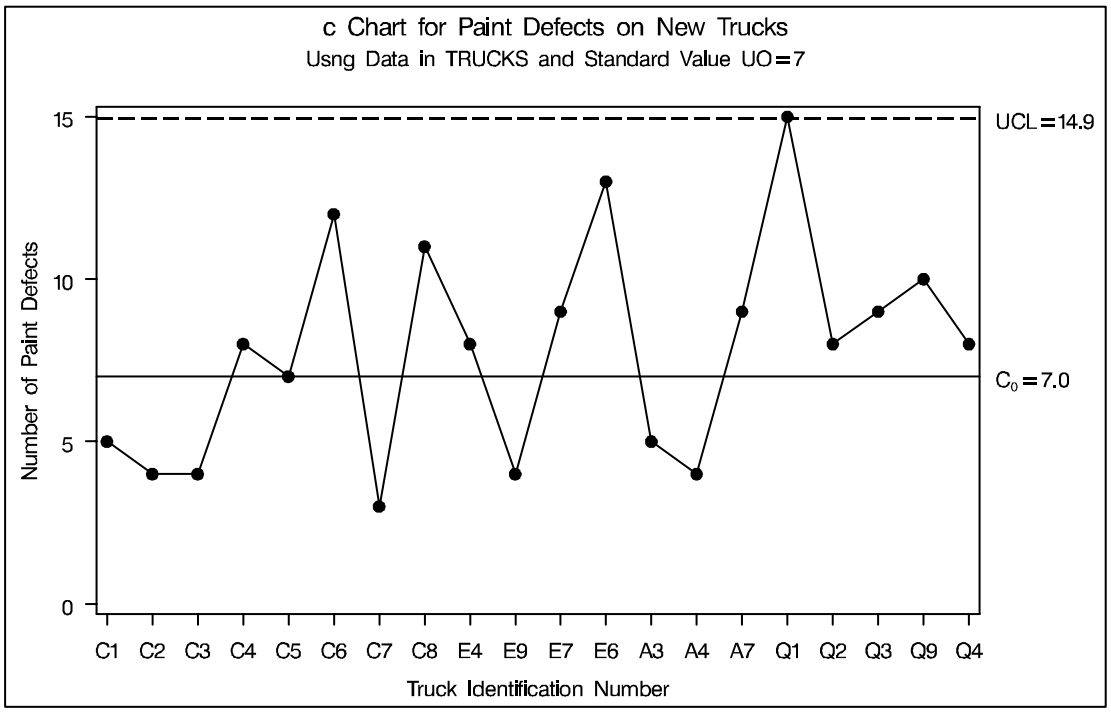
This example illustrates how you can create a c chart based on a known (standard) value u_0 for the expected number of nonconformities per unit.

A c chart is used to monitor the number of paint defects per truck. The defect counts are provided as values of the variable DEFECTS in the data set TRUCKS given on page 1306. Based on previous testing, it is known that $u_0 = 7$. The following statements create a c chart with control limits derived from this value:

```
symbol v = dot;
title 'c Chart for Paint Defects on New Trucks';
title2 'Usng Data in TRUCKS and Standard Value UO=7';
proc shewhart data=trucks;
   cchart defects*truckid / u0       = 7
                            csymbol  = c0
                            nolegend
                            nolimitslegend
                            nolimit0;
run;
```

The chart is shown in Output 40.2.1. The U0= option specifies u_0, and the CSYMBOL= option requests a label for the central line indicating that the line represents a standard value. The NOLEGEND option suppresses the legend for the subgroup sample size, and the NOLIMITSLEGEND option suppresses the legend for the control limits that appears by default in the upper right corner of the chart. The NOLIMIT0 option suppresses the display of the lower limit when it is equal to zero.

Output 40.2.1. A c Chart with Standard Value u_0



The number of paint defects on Truck Q1 exceeds the upper control limit, indicating that the process is out of control.

Alternatively, you can specify u_0 as the value of the variable _U_ in a LIMITS= data set, as follows:

```
data tlimits;
   length _subgrp_ _var_ _type_ $8;
   _U_      = 7;
   _subgrp_ = 'truckid';
   _var_    = 'defects';
   _limitn_ = 1;
   _type_   = 'STANDARD';

proc shewhart data=trucks limits=tlimits;
   cchart defects*truckid / csymbol=c0
                            nolegend
                            nolimitslegend
                            nolimit0;
run;
```

The chart produced by these statements is identical to the one in Output 40.2.1.

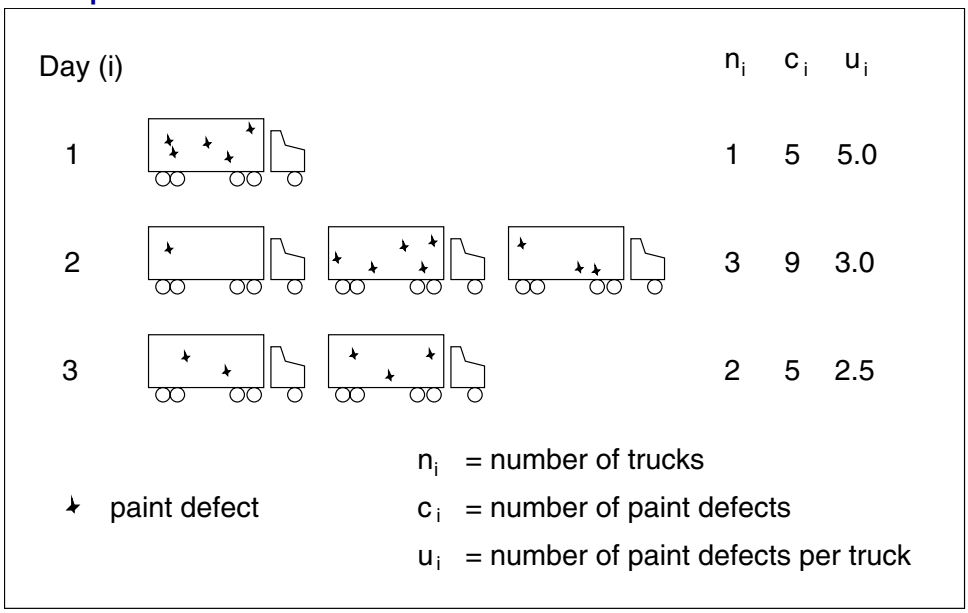
For further details, see "LIMITS= Data Set" on page 1333.

Example 40.3. Creating c Charts for Varying Numbers of Units

See SHWCEX3 in the SAS/QC Sample Library

In applications where the number of inspection units per subgroup is not equal to one, a u chart is typically used to analyze the number of nonconformities *per unit* (see Chapter 48, "UCHART Statement,"). However, as shown in this example, you can use the CCHART statement to create a c chart for this type of data.

Output 40.3.1. Difference between c Charts and u Charts



Output 40.3.1 illustrates a situation in which varying numbers of trucks are painted each day. Trucks painted on the same day are regarded as *subgroups*, and each truck is regarded as an *inspection unit*. The following statements create a SAS data set named TRUCKS4, which contains paint defects for trucks painted on 26 days:

```
data trucks4;
   input day defects ntrucks @@;
   label day='Day'
         defects='Number of Paint Defects';
   datalines;
 1   5  1    2   9  3
 3   5  2    4   9  2
 5  24  4    6  10  2
 7  15  3    8  17  3
 9  16  3   10  13  2
11  28  4   12  18  5
13   8  2   14   7  2
15   5  1   16  17  3
17   2  1   18  17  3
19  15  4   20  19  5
21   6  3   22  23  5
23  27  4   24   6  2
25  12  2   26  12  3
;
run;
```

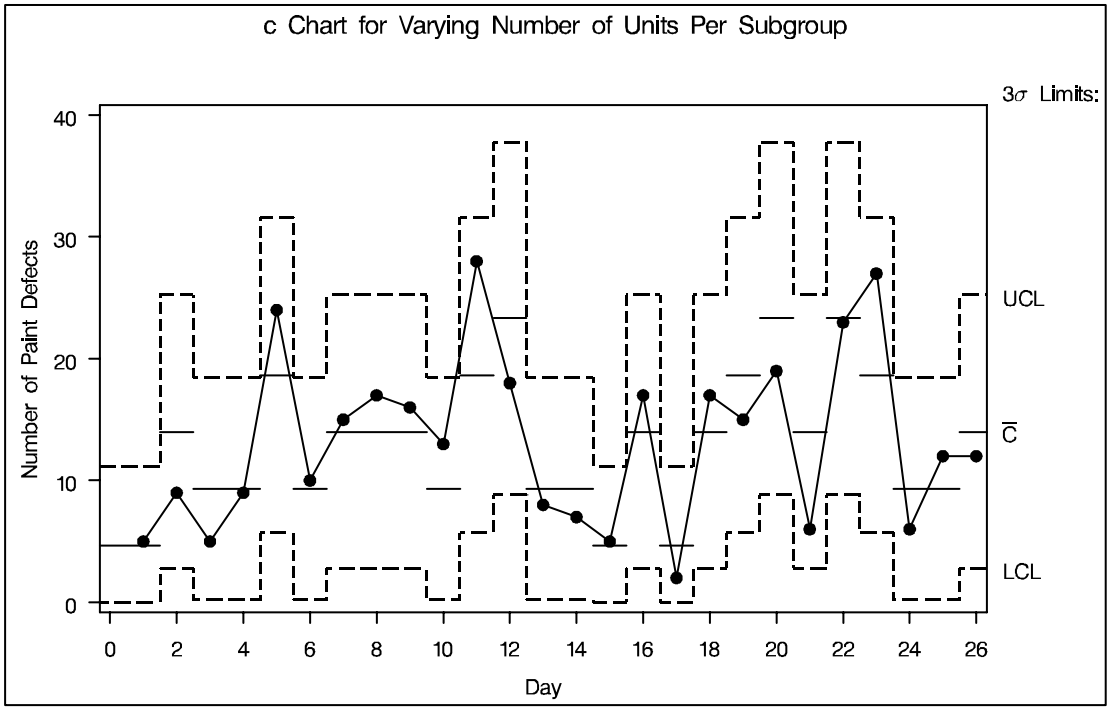
The variable DEFECTS provides the defect count (c_i) for the i^{th} day, and the variable NTRUCKS provides the number of inspection units (n_i). The following statements create a c chart for this data:

```
title 'c Chart for Varying Number of Units Per Subgroup';
proc shewhart data=trucks4;
   cchart defects*day / subgroupn = ntrucks
                        nolegend;
run;
```

The SUBGROUPN= option specifies the subgroup sample size variable NTRUCKS (in general, the values of this variable need not be integers). Alternatively, you can specify a fixed value with the SUBGROUPN= option. When this option is not specified, it is assumed that $n_i = 1$.

The chart is shown in Output 40.3.2. Note that the central line and the control limits vary with the number of inspection units.

Output 40.3.2. c Chart for Varying Number of Units



Chapter 41
IRCHART Statement

Chapter Contents

OVERVIEW 1347

GETTING STARTED 1348
- Creating Individual Measurements and Moving Range Charts 1348
- Saving Individual Measurements and Moving Ranges 1350
- Reading Individual Measurements and Moving Ranges 1351
- Saving Control Limits 1352
- Reading Preestablished Control Limits 1354
- Specifying the Computation of the Moving Range 1356

SYNTAX 1357
- Summary of Options 1358

DETAILS 1371
- Constructing Charts for Individual Measurements and Moving Ranges . . . 1371
- Output Data Sets 1373
- ODS Tables 1376
- Input Data Sets 1376
- Methods for Estimating the Standard Deviation 1379
- Interpreting Charts for Individual Measurements and Moving Ranges 1380
- Axis Labels 1381
- Missing Values 1381

EXAMPLES 1382
- Example 41.1. Applying Tests for Special Causes 1382
- Example 41.2. Specifying Standard Values for the Process Mean and Standard Deviation 1383
- Example 41.3. Displaying Distributional Plots in the Margin 1386

Chapter 41
IRCHART Statement

Overview

The IRCHART statement creates control charts for individual measurements and moving ranges. These charts are appropriate when only one measurement is available for each subgroup sample and when the measurements are independently and normally distributed.

You can use options in the IRCHART statement to

- compute control limits from the data based on a multiple of the standard error of the individual measurements and moving ranges or as probability limits
- tabulate individual measurements, moving ranges, and control limits
- save control limits in an output data set
- save individual measurements and moving ranges in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify a known (standard) process mean and standard deviation for computing control limits
- specify the number of consecutive measurements to use when computing the moving ranges
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

Getting Started

This section introduces the IRCHART statement with simple examples that illustrate commonly used options. Complete syntax for the IRCHART statement is presented in the "Syntax" section on page 1357, and advanced examples are given in the "Examples" section on page 1382.

Creating Individual Measurements and Moving Range Charts

See SHWIR1 in the SAS/QC Sample Library

An aeronautics company manufacturing jet engines measures the inner diameter of the forward face of each engine (in centimeters). The following statements create a SAS data set that contains the diameter measurements for 20 engines:

```
data jets;
   input engine diam @@;
   label engine = "Engine Number";
datalines;
 1 78.4   2 80.1   3 84.4   4 79.1   5 80.4
 6 83.5   7 73.8   8 83.5   9 75.0  10 76.8
11 70.5  12 80.3  13 82.4  14 79.4  15 86.4
16 90.5  17 77.7  18 82.5  19 79.9  20 83.2
;
run;
```

A partial listing of JETS is shown in Figure 41.1.

The Data Set JETS

engine	diam
1	78.4
2	80.1
3	84.4
4	79.1
.	.
.	.
.	.

Figure 41.1. Partial Listing of the Data Set JETS

Each observation contains the diameter measurement and identification number for a particular engine. The variable ENGINE identifies the sequence of engines and is referred to as the *subgroup-variable*.* The variable DIAM contains the measurements and is referred to as the *process variable* (or *process* for short).

*Technically, the data for individual measurements and moving range charts are not arranged in rational subgroups. The term *subgroup-variable* is used for consistency with other chart statements in the SHEWHART procedure, and it is convenient to think of the "subgroups" as consisting of single measurements.

Since the production rate is low, individual measurements and moving range charts are used to monitor the process. The following statements create the charts shown in Figure 41.2:

```
symbol h = .8;
title 'Individual Measurements and Moving Range Charts';
title2 'Jet Engine Diameters (cm)';
proc shewhart data=jets;
   irchart diam*engine;
run;
```

This example illustrates the basic form of the IRCHART statement. After the keyword IRCHART, you specify the *process* to analyze (in this case, DIAM), followed by an asterisk and the *subgroup-variable* (ENGINE).

The input data set is specified with the DATA= option in the PROC SHEWHART statement.

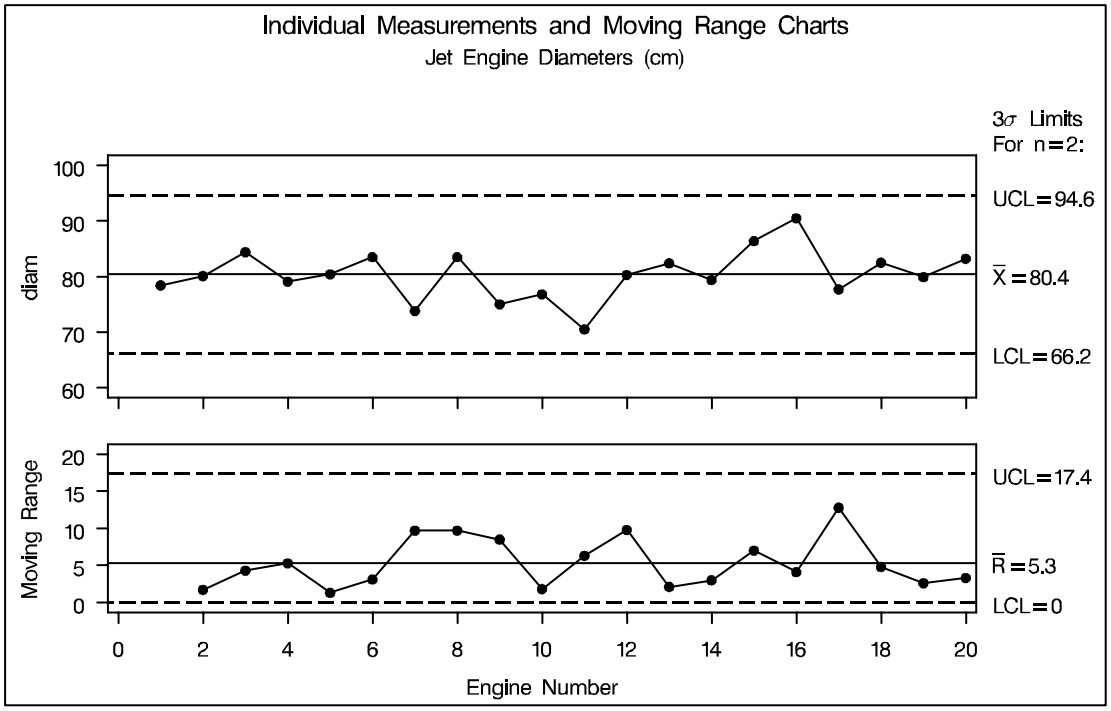


Figure 41.2. Individual Measurements and Moving Range Charts

Each point on the individual measurements chart indicates the inner diameter of a particular engine. Each point on the moving range chart indicates the range of the two most recent measurements. For instance, the moving range plotted for the second engine is $|78.4 - 80.1| = 1.7$. No moving range is plotted for the first engine. Since all of the individual measurements and moving ranges lie within the control limits, it can be concluded that the process is in statistical control.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given on page 1372. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1354.

Saving Individual Measurements and Moving Ranges

See SHWIR1 in the SAS/QC Sample Library

In this example, the IRCHART statement is used to create an output data set containing individual measurements and moving ranges. The following statements read the diameter measurements from the data set JETS (see page 1348) and create a data set named JETINFO:

```
proc shewhart data=jets;
   irchart diam*engine / outhistory = jetinfo
                         nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the charts, which would be identical to those in Figure 41.2. Options such as OUTHISTORY= and NOCHART are specified after the slash (/) in the IRCHART statement. A complete list of options is presented in the "Syntax" section on page 1357.

Figure 41.3 contains a partial listing of JETINFO.

Individual Measurements and Moving Ranges for Diameters

engine	diam	diam R
1	78.4	.
2	80.1	1.7
3	84.4	4.3
4	79.1	5.3
5	80.4	1.3
.	.	.
.	.	.
.	.	.

Figure 41.3. The Data Set JETINFO

The data set JETINFO contains one observation for each engine, and it includes three variables.

- ENGINE contains the subgroup index.
- DIAM contains the individual measurements.
- DIAMR contains the moving ranges.

Note that the variable containing the moving ranges is named by adding the suffix character *R* to the *process* DIAM specified in the IRCHART statement.

For more information, see "OUTHISTORY= Data Set" on page 1374.

Reading Individual Measurements and Moving Ranges

See SHWIR1 in the SAS/QC Sample Library

In some applications, both individual measurements and moving ranges may be provided. You can read this type of data set by specifying it with the HISTORY= option in the PROC SHEWHART statement. For example, the following statements read the data set JETINFO (see page 1350) and create the charts shown in Figure 41.4:

```
symbol h = .8;
title 'Individual Measurements and Moving Range Charts';
proc shewhart data=jets lineprinter;
   irchart diam*engine='*';
run;
```

Note that the charts are produced on a line printer since the LINEPRINTER option is specified in the PROC SHEWHART statement. * The asterisk (*) specified in single quotes after the *subgroup-variable* indicates the character used to plot points. This character must follow an equal sign.

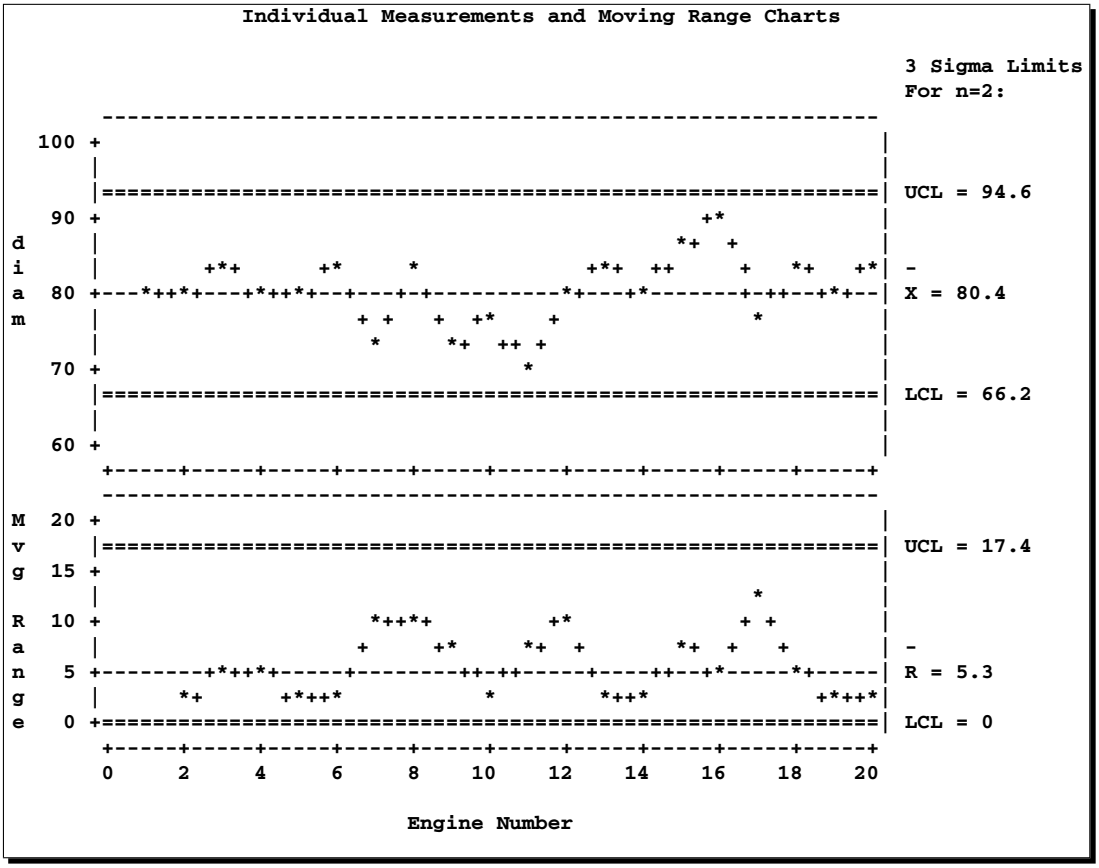


Figure 41.4. Charts from the Summary Data Set JETINFO

*In Release 6.12 and previous releases of SAS/QC software, the keyword GRAPHICS was required in the PROC SHEWHART statement to specify that the chart be created with a graphics device. In Version 7, you can specify the LINEPRINTER option to request line printer plots.

A HISTORY= data set used with the IRCHART statement must contain the following variables:

- subgroup variable
- individual measurements variable
- moving range variable

Furthermore, the name of the moving range variable must begin with the *process* name specified in the IRCHART statement and end with the special suffix character *R*. If the name does not follow this convention, you can use the RENAME option in the PROC SHEWHART statement to rename this variable for the duration of the procedure step (see page 1743). For more information, see "HISTORY= Data Set" on page 1377.

Saving Control Limits

See SHWIR1 in the SAS/QC Sample Library

You can save the control limits for individual measurements and moving range charts in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1354) or modify the limits with a DATA step program.

The following statements read the diameter measurements from the data set JETS (see page 1348) and save the control limits displayed in Figure 41.2 in a data set named JETLIM:

```
proc shewhart data=jets;
   irchart diam*engine / outlimits = jetlim
                         nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the charts. The data set JETLIM is listed in Figure 41.5.

Control Limits for Diameters

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLI_
diam	engine	ESTIMATE	2	.002699796	3	66.2290

MEAN	_UCLI_	_LCLR_	_R_	_UCLR_	_STDDEV_
80.39	94.5510	0	5.32632	17.3986	4.72032

Figure 41.5. The Data Set JETLIM Containing Control Limit Information

The data set JETLIM contains one observation with the limits for *process* DIAM. The variables _LCLI_ and _UCLI_ contain the control limits for the individual measurements, and the variable _MEAN_ contains the central line. The variables _LCLR_

and _UCLR_ contain the control limits for the moving ranges, and the variable _R_ contains the central line. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the number of consecutive measurements used to compute the moving ranges, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values. For more information, see "OUTLIMITS= Data Set" on page 1373.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=jets;
   irchart diam*engine / outtable=jtable
                         nochart;
run;
```

The data set JTABLE is listed in Figure 41.6.

```
         Summary Statistics and Control Limit Information


        _ _                                                     _
        S L                                                     E
                                       _
      e I I    _      _    _      _    E _   _             _    X
 _    n G M    L      S    M      U    X L   S             U    L
 V    g M I    C      U    E      C    L C   U             C    I
 A    i A T    L      B    A      L    I L   B     _       L    M
 R    n S N    I      I    N      I    M R   R     R       R    R
 _    e _ _    _      _    _      _    _ _   _     _       _    _

diam  1 3 2 66.2290 78.4 80.39 94.5510   0   .  5.32632 17.3986
diam  2 3 2 66.2290 80.1 80.39 94.5510   0  1.7 5.32632 17.3986
diam  3 3 2 66.2290 84.4 80.39 94.5510   0  4.3 5.32632 17.3986
diam  4 3 2 66.2290 79.1 80.39 94.5510   0  5.3 5.32632 17.3986
diam  5 3 2 66.2290 80.4 80.39 94.5510   0  1.3 5.32632 17.3986
diam  6 3 2 66.2290 83.5 80.39 94.5510   0  3.1 5.32632 17.3986
diam  7 3 2 66.2290 73.8 80.39 94.5510   0  9.7 5.32632 17.3986
diam  8 3 2 66.2290 83.5 80.39 94.5510   0  9.7 5.32632 17.3986
diam  9 3 2 66.2290 75.0 80.39 94.5510   0  8.5 5.32632 17.3986
diam 10 3 2 66.2290 76.8 80.39 94.5510   0  1.8 5.32632 17.3986
diam 11 3 2 66.2290 70.5 80.39 94.5510   0  6.3 5.32632 17.3986
diam 12 3 2 66.2290 80.3 80.39 94.5510   0  9.8 5.32632 17.3986
diam 13 3 2 66.2290 82.4 80.39 94.5510   0  2.1 5.32632 17.3986
diam 14 3 2 66.2290 79.4 80.39 94.5510   0  3.0 5.32632 17.3986
diam 15 3 2 66.2290 86.4 80.39 94.5510   0  7.0 5.32632 17.3986
diam 16 3 2 66.2290 90.5 80.39 94.5510   0  4.1 5.32632 17.3986
diam 17 3 2 66.2290 77.7 80.39 94.5510   0 12.8 5.32632 17.3986
diam 18 3 2 66.2290 82.5 80.39 94.5510   0  4.8 5.32632 17.3986
diam 19 3 2 66.2290 79.9 80.39 94.5510   0  2.6 5.32632 17.3986
diam 20 3 2 66.2290 83.2 80.39 94.5510   0  3.3 5.32632 17.3986
```

Figure 41.6. The Data Set JTABLE

This data set contains one observation for each subgroup. The variables _SUBI_ and _SUBR_ contain the individual measurements and moving ranges. The variables

LCLI and _UCLI_ contain the lower and upper control limits for the individual measurements chart, and the variables _LCLR_ and _UCLR_ contain the lower and upper control limits for the moving range chart. The variable _MEAN_ contains the central line of the individual measurements chart, and the variable _R_ contains the central line of the moving range chart. The variables _VAR_ and ENGINE contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1374.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read JTABLE and display charts (not shown here) identical to those in Figure 41.2:

```
title 'Individual Measurements and Moving Range Control Charts';
title2 'Jet Engine Diameters (cm)';
proc shewhart table=jtable;
   irchart diam*engine;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see Chapter 56, "Specialized Control Charts,").

For more information, see "TABLE= Data Set" on page 1378.

Reading Preestablished Control Limits

See SHWIR1 in the SAS/QC Sample Library

In the previous example, the OUTLIMITS= data set JETLIM saved control limits computed from the measurements in JETS. This example shows how these limits can be applied to data for an additional 20 jet engines provided in the following data set:

```
data jets2;
   input engine diam @@;
   label diam   = "Inner Diameter (cm)"
         engine = "Engine Number";
datalines;
21 81.8  22 87.5  23 80.0  24 89.3  25 83.9
26 76.3  27 75.8  28 82.4  29 82.6  30 77.7
31 79.3  32 81.4  33 76.8  34 75.9  35 86.3
36 77.4  37 80.9  38 87.1  39 85.7  40 73.3
;
run;
```

The following statements create individual measurements and moving range charts for the data in JETS2 using the control limits in JETLIM:

```
symbol h = .8;
title 'Individual Measurements and Moving Range Control Charts';
proc shewhart data=jets2 limits=jetlim;
   irchart diam*engine;
run;
```

The charts are shown in Figure 41.7. The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default,* this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name DIAM
- the value of _SUBGRP_ matches the *subgroup-variable* name ENGINE

The charts indicate that the process is in control, since all the individual measurements and moving ranges lie within their respective control limits.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1376 for details concerning the variables that you must provide.

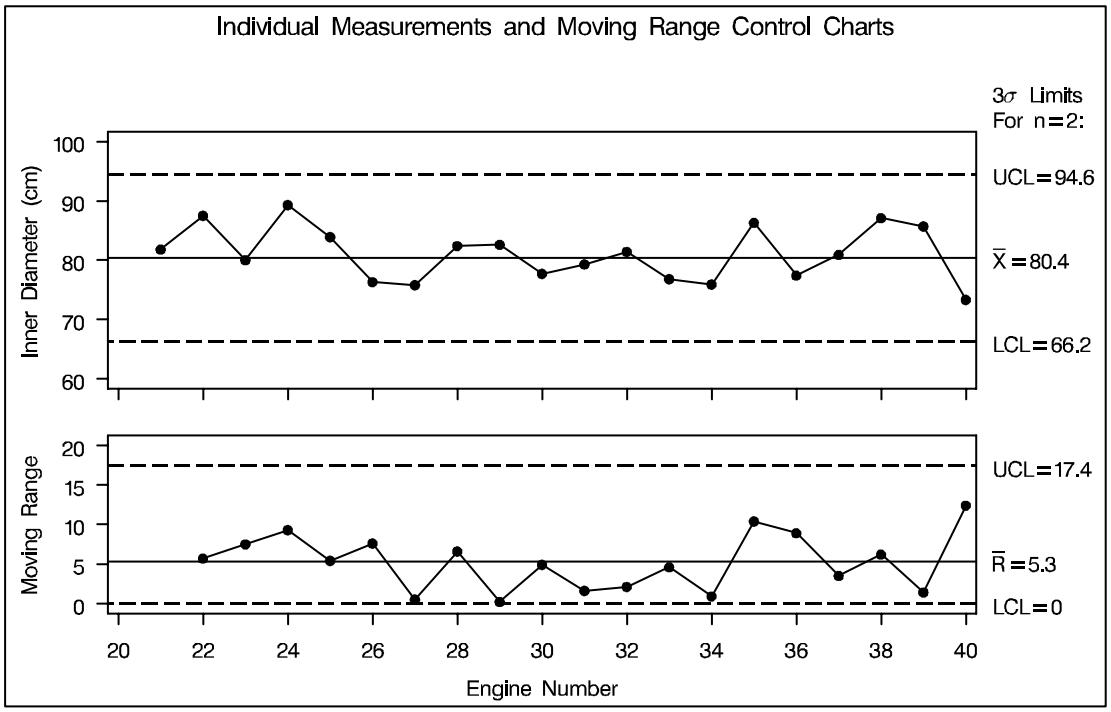


Figure 41.7. Charts for Second Set of Engine Noise Levels

*In Release 6.09 and in earlier releases, it is also necessary to specify the READLIMITS option to read control limits from a LIMITS= data set.

Specifying the Computation of the Moving Range

See SHWIR1 in the SAS/QC Sample Library

By default, the IRCHART statement uses two consecutive measurements to calculate moving ranges. However, you can specify a different number of measurements to use, as illustrated by the following statements:

```
symbol h = .8;
title 'Specifying the Computation of the Moving Range';
proc shewhart data=jets;
   irchart diam*engine / limitn=3;
run;
```

The LIMITN= option specifies the number of consecutive measurements used to compute the moving ranges. The resulting charts are shown in Figure 41.8.

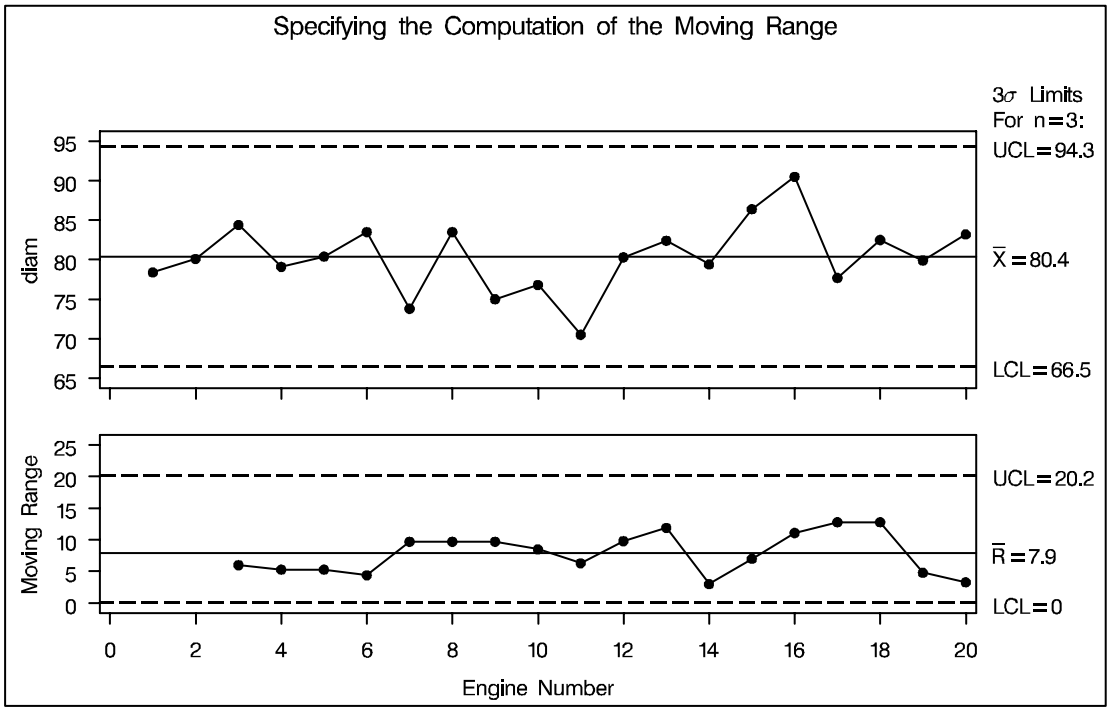


Figure 41.8. Computing Moving Ranges from Three Consecutive Measurements

Note that the LIMITN= value is displayed in the legend above the control limit labels. The charts indicate that the process is in control, since all the points lie within the control limits.

Syntax

The basic syntax for the IRCHART statement is as follows:

IRCHART *process******subgroup-variable* **;**

The general form of this syntax is as follows:

IRCHART (*processes***)***subgroup-variable* <**(***block-variables* **)** >
< =*symbol-variable* | =*'character'* > < **/** *options* >**;**

You can use any number of IRCHART statements in the SHEWHART procedure. The components of the IRCHART statement are described as follows.

process
processes
identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the individual measurements. For an example, see "Creating Individual Measurements and Moving Range Charts" on page 1348.
- If individual measurements and moving ranges are read from a HISTORY= data set, *process* must be the name of the variable containing the individual measurements as well as the prefix of the variable containing the moving ranges in the HISTORY= data set. For an example, see "Saving Individual Measurements and Moving Ranges" on page 1350.
- If individual measurements, moving ranges, and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1352.

A *process* is required. If you specify more than one *process*, enclose the list in parentheses. For example, the following statements request distinct individual measurements and moving range charts for WEIGHT, LENGTH, and WIDTH:

```
proc shewhart data=measures;
   irchart (weight length width)*day;
run;
```

subgroup-variable
is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding IRCHART statement, DAY is the subgroup variable. Note that each "subgroup" consists of a single observation. For details, see "Subgroup Variables" on page 1771.

block-variables
are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in

the legend. See "Displaying Stratification in Blocks of Observations" on page 1932 for an example.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the individual measurements and moving ranges.

- If you produce a chart on a line printer, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce a chart on a graphics device, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOLn statements. See "Displaying Stratification in Levels of a Classification Variable" on page 1931 for an example.

character
specifies a plotting character for charts produced on line printers. For example, the following statements create charts using an asterisk (*) to plot the points:

```
proc shewhart data=values;
   irchart weight*day='*';
run;
```

options
enhance the appearance of the charts, request additional analyses, save results in data sets, and so on. The "Summary of Options" section, which follows, lists all options by function. Chapter 53, "Dictionary of Options," describes each option in detail.

Summary of Options

The following tables list the IRCHART statement options by function. For complete descriptions, see Chapter 53, "Dictionary of Options."

Table 41.1. Tabulation Options

TABLE	creates a basic table of individual measurements, moving ranges, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUT, and TABLETESTS
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with a column indicating which tests for special causes are positive

Note that specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.

Table 41.2. Options for Specifying Tests for Special Causes

NO3SIGMACHECK	allows tests to be applied with control limits other than 3σ limits
TESTS=*value-list*| *customized-pattern-list*	specifies tests for special causes for the individual measurements chart
TEST2RUN=*n*	specifies length of pattern for Test 2
TEST3RUN=*n*	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL='*label*'| (*variable*)|*keyword*	provides labels for points where test is positive
TESTLABEL*n*='*label*'	specifies label for n^{th} test for special causes
TESTNMETHOD= STANDARDIZE	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
TESTRESET=*variable*	allows tests for special causes to be reset for the individual measurements chart
ZONELABELS	adds labels A, B, and C to zone lines on individual measurements chart
ZONES	adds lines to individual measurements chart delineating zones A, B, and C
ZONEVALPOS=*n*	specifies position of ZONEVALUES labels
ZONEVALUES	labels zone lines with their values

Table 41.3. Graphical Options for Displaying Tests for Special Causes

CTESTLABBOX=*color*	specifies color for boxes enclosing labels indicating points where test is positive
CTESTSYMBOL=*color*	specifies color for symbol used to plot points where test is positive
CTESTS=*color*| *test-color-list*	specifies color for labels used to identify points where test is positive
CZONES=*color*	specifies color for lines and labels delineating zones A, B, and C
LTESTS=*linetype*	specifies type of line connecting points where test is positive
LZONES=*linetype*	specifies line type for lines delineating zones A, B, and C
TESTFONT=*font*	specifies software font for labels at points where test is positive
TESTHEIGHT=*value*	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=*symbol*	specifies plot symbol for points where test is positive
TESTSYMBOLHT=*value*	specifies symbol height for points where test is positive
WTESTS=*n*	specifies width of line connecting points where test is positive

Table 41.4. Line Printer Options for Displaying Tests for Special Causes

TESTCHAR='*character*'	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
ZONECHAR='*character*'	specifies character for lines that delineate zones for tests for special causes

Table 41.5. Clipping Options

CCLIP=*color*	specifies color for plot symbol for clipped points
CLIPCHAR='*character*'	specifies plot character for clipped points
CLIPFACTOR=*value*	determines extent to which extreme points are clipped
CLIPLEGEND='*string*'	specifies text for clipping legend
CLIPLEGPOS=*keyword*	specifies position of clipping legend
CLIPSUBCHAR='*character*'	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=*symbol*	specifies plot symbol for clipped points
CLIPSYMBOLHT=*value*	specifies symbol marker height for clipped points

Table 41.6. Options for Plotting and Labeling Points

ALLLABEL=VALUE|(*variable*)	labels every point on individual measurements chart
ALLLABEL2=VALUE|(*variable*)	labels every point on moving range chart
CLABEL=*color*	specifies color for labels
CCONNECT=*color*	specifies color for line segments that connect points on chart
CFRAMELAB=*color*	specifies fill color for frame around labeled points
CNEEDLES=*color*	specifies color for needles that connect points to central line
CONNECTCHAR='*character*'	specifies character used to form line segments that connect points on chart
COUT=*color*	specifies color for portions of line segments that connect points outside control limits
COUTFILL=*color*	specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=*angle*	specifies angle at which labels are drawn
LABELFONT=*font*	specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=*value*	specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on chart
OUTLABEL=VALUE|(*variable*)	labels points outside control limits on individual measurements chart
OUTLABEL2=VALUE|(*variable*)	labels points outside control limits on moving range chart
SYMBOLCHARS='*characters*'	specifies characters indicating *symbol-variable*
SYMBOLLEGEND=NONE|*name*	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=*keyword*	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=*n*	specifies width of needles

Table 41.7. Reference Line Options

CHREF=*color*	specifies color for lines requested by the HREF= and HREF2= options
CVREF=*color*	specifies color for lines requested by the VREF= and VREF2= options
HREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on individual measurements chart
HREF2=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on moving range chart
HREFCHAR='*character*'	specifies line character for HREF= and HREF2= lines
HREFDATA= *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on individual measurements chart
HREF2DATA= *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on moving range chart
HREFLABELS= (*'label1'...'labeln'*)	specifies labels for HREF= lines
HREF2LABELS= (*'label1'...'labeln'*)	specifies labels for HREF2= lines
HREFLABPOS=*n*	specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=*linetype*	specifies line type for HREF= and HREF2= lines
LVREF=*linetype*	specifies line type for VREF= and VREF2= lines
NOBYREF	specifies that reference line information in a data set is to be applied uniformly to charts created for all BY groups
VREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to vertical axis on individual measurements chart
VREF2=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to vertical axis on moving range chart
VREFCHAR='*character*'	specifies line character for VREF= and VREF2= lines
VREFLABELS= *'label1'...'labeln'*	specifies labels for VREF= lines
VREF2LABELS= *'label1'...'labeln'*	specifies labels for VREF2= lines
VREFLABPOS=*n*	specifies position of VREFLABELS= and VREF2LABELS= labels

Table 41.8. Block Variable Legend Options

BLOCKLABELPOS= *keyword*	specifies position of label for *block-variable* legend
BLOCKLABTYPE= *n*|*keyword*	specifies text size of *block-variable* legend
BLOCKPOS=*n*	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=*color-list*	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=*variable*| (*variables*)	specifies one or more variables whose values are colors for filling background of *block-variable* legend

Table 41.9. Process Mean and Standard Deviation Options

MU0=*value*	specifies known value μ_0 for process mean μ
SIGMA0=*value*	specifies known value σ_0 for process standard deviation σ
SMETHOD=*keyword*	specifies method for estimating process standard deviation σ
TYPE=*keyword*	identifies whether parameters are estimates or standard values and specifies value of _TYPE_ in OUTLIMITS= data set

Table 41.10. Phase Options

CPHASELEG=*color*	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE='*string*'	specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=*value*| *keyword*	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES= ALL| '*label1*'...'*labeln*'	specifies *phases* to be read from an input data set

Table 41.11. Axis and Axis Label Options

CAXIS=*color*	specifies color for axis lines and tick marks
CFRAME=*color*|(*color-list*)	specifies fill colors for frame for plot area
CTEXT=*color*	specifies color for tick mark values and axis labels
HAXIS=*values*|AXISn	specifies major tick mark values for horizontal axis
HEIGHT=*value*	specifies height of axis label and axis legend text
HMINOR=*n*	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=*value*	specifies length of offset at both ends of horizontal axis
INTSTART=*value*	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	suppresses vertical axis truncation at zero applied by default to moving range chart
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=*n*	specifies thinning factor for tick mark labels on horizontal axis
SPLIT='*character*'	specifies splitting character for axis labels
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=*values*|AXISn	specifies major tick mark values for vertical axis of individual measurements chart
VAXIS2=*values*|AXISn	specifies major tick mark values for vertical axis of moving range chart
VFORMAT=*format*	specifies format for primary vertical axis tick mark labels
VFORMAT2=*format*	specifies format for secondary vertical axis tick mark labels
VMINOR=*n*	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=*value*	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis for primary chart
VZERO2	forces origin to be included in vertical axis for secondary chart
WAXIS=*n*	specifies width of axis lines

Table 41.12. Options for Interactive Control Charts

HTML=(*variable*)	specifies a variable whose values are URLs to be associated with subgroups
HTML2=(*variable*)	specifies variable whose values are URLs to be associated with subgroups on secondary chart
HTML_LEGEND=(*variable*)	specifies variable whose values are URLs to be associated with symbols in the symbol legend
WEBOUT=*SAS-data-set*	creates an OUTTABLE= data set with additional graphics coordinate data

Table 41.13. Options for Displaying Control Limits

CINFILL=*color*	specifies color for area inside control limits
CLIMITS=*color*	specifies color of control limits, central line, and related labels
LCLLABEL='*label*'	specifies label for lower control limit on individual measurements chart
LCLLABEL2='*label*'	specifies label for lower control limit on moving range chart
LIMLABSUBCHAR='*character*'	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=*linetype*	specifies line type for control limits
NDECIMAL=*n*	specifies number of digits to right of decimal place in default labels for control limits and central line on individual measurements chart
NDECIMAL2=*n*	specifies number of digits to right of decimal place in default labels for control limits and central line on moving range chart
NOCTL	suppresses display of central line on individual measurements chart
NOCTL2	suppresses display of central line on moving range chart
NOLCL	suppresses display of lower control limit on individual measurements chart
NOLCL2	suppresses display of lower control limit on moving range chart
NOLIMITLABEL	suppresses labels for control limits and central lines
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOLIMIT0	suppresses display of zero lower control limit on moving range chart
NOUCL	suppresses display of upper control limit on individual measurements chart
NOUCL2	suppresses display of upper control limit on moving range chart
RSYMBOL='*string*'|*keyword*	specifies label for central line on R chart
UCLLABEL='*string*'	specifies label for upper control limit on individual measurements chart
UCLLABEL2='*string*'	specifies label for upper control limit on moving range chart
WLIMITS=*n*	specifies width for control limits and central line
XSYMBOL='*string*'|*keyword*	specifies label for central line on individual measurements chart

Table 41.14. Options for Specifying Control Limits

ALPHA=*value*	requests probability limits for control charts
LIMITN=*n*	specifies number of consecutive measurements used to compute moving ranges
MRRESTART	restarts the moving range computation at missing values
NOREADLIMITS	computes control limits for each *process* from the data rather than from a LIMITS= data set (Release 6.10 and later releases)
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEXES=ALL| '*label1*'...'*labeln*'	reads multiple sets of control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (Release 6.09 and earlier releases)
SIGMAS=*k*	specifies width of control limits in terms of multiple k of standard error of plotted statistic

Table 41.15. Plot Layout Options

BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts only when exceptions occur
INTERVAL=*keyword*	specifies natural time interval between consecutive subgroup positions when time, date, or datetime format is used
MAXPANELS=*n*	specifies maximum number of pages or screens for chart
NOCHART	suppresses creation of charts
NOCHART2	suppresses creation of moving range chart
NOFRAME	suppresses frame for plot area
NPANELPOS=*n*	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
SEPARATE	displays individual measurements and moving range charts on separate screens or pages
TOTPANELS=*n*	specifies number of pages or screens to be used to display chart
YPCT1=*value*	specifies length of vertical axis on individual measurements chart as a percentage of sum of lengths of vertical axes for individual measurements and moving range charts
ZEROSTD	displays individual measurements and moving range charts regardless of whether $\hat{\sigma} = 0$

Table 41.16. Graphical Enhancement Options

ANNOTATE= *SAS-data-set*	specifies annotate data set that adds features to individual measurements chart
ANNOTATE2= *SAS-data-set*	specifies annotate data set that adds features to moving range chart
DESCRIPTION='*string*'	specifies string that appears in the description field of the PROC GREPLAY master menu for individual measurements chart
DESCRIPTION2='*string*'	specifies string that appears in the description field of the PROC GREPLAY master menu for moving range chart
FONT=*font*	specifies software font for labels and legends on charts
LTMARGIN=*value*	specifies width of left margin area for plot requested with LTMPLOT= option
LTMPLOT=*keyword*	requests univariate plot in left margin
NAME='*string*'	specifies name that appears in the name field of the PROC GREPLAY master menu for individual measurements chart
NAME2='*string*'	specifies name that appears in the name field of the PROC GREPLAY master menu for moving range chart
PAGENUM='*string*'	specifies the form of the label used in pagination
PAGENUMPOS= *keyword*	specifies the position of the page number requested with the PAGENUM= option
RTMARGIN=*value*	specifies width of right margin area for plot requested with RTMPLOT= option
RTMPLOT=*keyword*	requests univariate plot in right margin

Table 41.17. Grid Options

CGRID=*color*	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=*linetype*	specifies line type for grid requested with the ENDGRID option
LGRID=*linetype*	specifies line type for grid requested with the GRID option
WGRID=*n*	specifies width of grid lines

Table 41.18. Specification Limit Options

CIINDICES (ALPHA=*value* TYPE=*keyword*)	specifies α value and type for computing capability index confidence limits
LSL=*value-list*	specifies list of lower specification limits
TARGET=*value-list*	specifies list of target values
USL=*value-list*	specifies list of upper specification limits

Table 41.19. Input Data Set Options

MISSBREAK	specifies that observations with missing values are not to be processed

Table 41.20. Output Data Set Options

OUTHISTORY= *SAS-data-set*	creates output data set containing individual measurements and moving ranges
OUTINDEX='*string*'	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS= *SAS-data-set*	creates output data set containing control limits
OUTTABLE= *SAS-data-set*	creates output data set containing individual measurements, moving ranges, and control limits

Table 41.21. Star Options

CSTARCIRCLES=*color*	specifies color for STARCIRCLES= circles
CSTARFILL=*color*| *(variable)*	specifies color for filling stars
CSTAROUT=*color*	specifies outline color for stars exceeding inner or outer circles
CSTARS=*color*| *(variable)*	specifies color for outlines of stars
LSTARCIRCLES= *linetypes*	specifies line types for STARCIRCLES= circles
LSTARS=*linetype*|*(variable)*	specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=*value*	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=*value-list*	specifies reference circles for stars
STARINRADIUS=*value*	specifies inner radius of stars
STARLABEL=*keyword*	specifies vertices to be labeled
STARLEGEND=*keyword*	specifies style of legend for star vertices
STARLEGENDLAB='*label*'	specifies label for STARLEGEND= legend
STAROUTRADIUS=*value*	specifies outer radius of stars
STARSPEC=*value*| *SAS-data-set*	specifies method used to standardize vertex variables
STARSTART=*value*	specifies angle for first vertex
STARTYPE=*keyword*	specifies graphical style of star
STARVERTICES= *variable*|*(variables)*	superimposes star at each point on individual measurements chart
WSTARCIRCLES=*n*	specifies width of STARCIRCLES= circles
WSTARS=*n*	specifies width of STARVERTICES= stars

Table 41.22. Overlay Options

CCOVERLAY=*color-list*	specifies colors for primary chart overlay line segments
CCOVERLAY2=*color-list*	specifies colors for secondary chart overlay line segments
COVERLAY=*color-list*	specifies colors for primary chart overlay plots
COVERLAY2=*color-list*	specifies colors for secondary chart overlay plots
COVERLAYCLIP=*color*	specifies color for clipped points on overlays
LOVERLAY=*linetypes*	specifies line types for primary chart overlay line segments
LOVERLAY2=*linetypes*	specifies line types for secondary chart overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=*variable-list*	specifies variables to overlay on primary chart
OVERLAY2=*variable-list*	specifies variables to overlay on secondary chart
OVERLAY2HTML=*variable-list*	specifies URLS to associate with secondary chart overlay points
OVERLAY2ID=*variable-list*	specifies labels for secondary chart overlay points
OVERLAY2SYM=*symbol-list*	specifies symbols for secondary chart overlays
OVERLAY2SYMHT=*value-list*	specifies symbol heights for secondary chart overlays
OVERLAYCLIPSYM=*symbol*	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=*value*	specifies symbol height for clipped points on overlays
OVERLAYHTML=*variable-list*	specifies URLS to associate with primary chart overlay points
OVERLAYID=*variable-list*	specifies labels for primary chart overlay points
OVERLAYLEGLAB='*label*'	specifies label for overlay legend
OVERLAYSYM=*symbol-list*	specifies symbols for primary chart overlays
OVERLAYSYMHT=*value-list*	specifies symbol heights for primary chart overlays
WOVERLAY=*value-list*	specifies widths of primary chart overlay line segments
WOVERLAY2=*value-list*	specifies widths of secondary chart overlay line segments

Details

Constructing Charts for Individual Measurements and Moving Ranges

The following notation is used in this section:

μ	process mean (expected value of the population of measurements)
σ	process standard deviation (standard deviation of the population of measurements)
X_i	the i^{th} individual measurement
$\bar{X}$	mean of the individual measurements, computed as $(X_1 + \cdots + X_N)/N$, where N is the number of individual measurements
n	number of consecutive measurements used to calculate the moving ranges (by default, $n = 2$)
R_i	moving range computed for the i^{th} subgroup (corresponding to the i^{th} individual measurement). If $i < n$, then R_i is assigned a missing value. Otherwise, $$R_i = \max(X_i, X_{i-1}, ..., X_{i-n+1}) - \min(X_i, X_{i-1}, ..., X_{i-n+1})$$ This formula assumes that $X_i, X_{i-1}, ..., X_{i-n+1}$ are nonmissing.
$\bar{R}$	average of the nonmissing moving ranges, computed as $$\frac{R_n + R_{n+1} \cdots + R_N}{N + 1 - n}$$
$d_2(n)$	expected value of the range of n independent normally distributed variables with unit standard deviation
$d_3(n)$	standard error of the range of n independent observations from a normal population with unit standard deviation
z_p	$100p^{\text{th}}$ percentile $(0 < p < 1)$ of the standard normal distribution
$D_p(n)$	$100p^{\text{th}}$ percentile $(0 < p < 1)$ of the distribution of the range of n independent observations from a normal population with unit standard deviation

Plotted Points

Each point on an individual measurements chart, indicates the value of a measurement (X_i).

Each point on a moving range chart indicates the value of a moving range (R_i). With $n = 2$, for example, if the first three measurements are 3.4, 3.7, and 3.6, the first moving range is missing, the second moving range is $|3.7 - 3.4| = 0.3$, and the third moving range is $|3.6 - 3.7| = 0.1$.

Central Lines

By default, the central line on an individual measurements chart indicates an estimate for μ, which is computed as $\bar{X}$. If you specify a known value (μ_0) for μ, the central line indicates the value of μ_0.

The central line on a moving range chart indicates an estimate for the expected moving range, computed as $d_2(n)\hat{\sigma}$ where $\hat{\sigma} = \bar{R}/d_2(n)$. If you specify a known value ($\hat{\sigma}_0$) for σ, the central line indicates the value of $d_2(n)\sigma_0$.

Control Limits

You can compute the limits

- as a specified multiple (k) of the standard errors of X_i and R_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that X_i or R_i exceeds the limits

The following table provides the formulas for the limits:

Table 41.23. Limits for Individual Measurements and Moving Range Charts

Control Limits	
Individual Measurements Chart	LCL = lower control limit = $\bar{X} - k\hat{\sigma}$ UCL = upper control limit = $\bar{X} + k\hat{\sigma}$
Moving Range Chart	LCL = lower control limit = $\max(d_2(n)\hat{\sigma} - kd_3(n)\hat{\sigma}, 0)$ UCL = upper control limit = $d_2(n)\hat{\sigma} + kd_3(n)\hat{\sigma}$

Probability Limits	
Individual Measurements Chart	LCL = lower control limit = $\bar{X} - z_{\alpha/2}\hat{\sigma}$ UCL = upper control limit = $\bar{X} + z_{\alpha/2}\hat{\sigma}$
Moving Range Chart	LCL = lower control limit = $D_{\alpha/2}(n)\hat{\sigma}$ UCL = upper control limit = $D_{1-\alpha/2}(n)\hat{\sigma}$

The formulas assume that the measurements are normally distributed. Note that the probability limits for the moving range are asymmetric about the central line. If standard values μ_0 and σ_0 are available for μ and σ, replace $\bar{X}$ with μ_0 and $\hat{\sigma}$ with σ_0 in Table 41.23.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify n with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify μ_0 with the MU0= option or with the variable _MEAN_ in the LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in the LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables can be saved:

Table 41.24. OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
_CP_	capability index C_p
CPK	capability index C_{pk}
CPL	capability index CPL
CPM	capability index C_{pm}
CPU	capability index CPU
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLI	lower control limit for individual measurements
LCLR	lower control limit for moving ranges
LIMITN	number of consecutive measurements used to compute moving ranges
LSL	lower specification limit
MEAN	process mean
_R_	value of central line on moving range chart
SIGMAS	multiple (k) of standard error of individual measurement or moving range
STDDEV	process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*subgroup-variable* specified in the IRCHART statement
TARGET	target value
TYPE	type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLI	upper control limit for individual measurements
UCLR	upper control limit for moving ranges range
USL	upper specification limit
VAR	*process* specified in the IRCHART statement

Notes:

1. If the limits are defined in terms of a multiple k of the standard errors of X_i and R_i, the value of _ALPHA_ is computed as $\alpha = 2(1 - \Phi(k))$, where $\Phi(\cdot)$ is the standard normal distribution function.
2. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = \Phi^{-1}(1 - \alpha/2)$, where Φ^{-1} is the inverse standard normal distribution function.
3. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you

provide a target value with the TARGET= option. See "Capability Indices" on page 1774 for computational details.

4. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the IRCHART statement. For an example, see "Saving Control Limits" on page 1352.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves individual measurements and moving ranges. The following variables are saved:

- the *subgroup-variable*
- an individual measurements variable named by *process*
- a moving range variable named by *process* suffixed with *R*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

A variable containing the moving ranges is created for each *process* specified in the IRCHART statement. For example, consider the following statements:

```
proc shewhart data=steel;
   irchart (width diameter)*lot / outhistory=summary;
run;
```

The data set SUMMARY contains variables named LOT, WIDTH, WIDTHR, DIAMETER, and DIAMTERR.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Individual Measurements and Moving Ranges" on page 1350.

OUTTABLE= Data Set

The OUTTABLE= data set saves individual measurements, moving ranges, control limits, and related information. The following variables are saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on individual measurements chart
EXLIMR	control limit exceeded on moving range chart
LCLI	lower control limit for individual measurements
LCLR	lower control limit for moving range
LIMITN	number of consecutive measurements used to compute moving ranges
MEAN	process mean
_R_	average range
SIGMAS	multiple (k) of the standard error associated with control limits
subgroup	values of the subgroup variable
SUBI	individual measurement
SUBR	moving range
TESTS	tests for special causes signaled on individual measurements chart
UCLI	upper control limit for individual measurements
UCLR	upper control limit for moving range
VAR	*process* specified in the IRCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved, depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The k^{th} character of a value of _TESTS_ is k if Test k is positive at that subgroup. For example, if you request all eight tests and Tests 2 and 8 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, an 8 for the eighth character, and blanks for the other six characters.
3. The variables _EXLIM_, _EXLIMR_, and _TESTS_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1352.

ODS Tables

The following table summarizes the ODS tables that you can request with the IRCHART statement.

Table 41.25. ODS Tables Produced with the IRCHART Statement

Table Name	Description	Options
IRCHART	individual measurement and moving range chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

Input Data Sets

DATA= Data Set

You can read individual measurements from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the IRCHART statement must be a SAS variable in the data set. This variable provides measurements of items indexed by the *subgroup-variable*. The *subgroup-variable*, which is specified in the IRCHART statement, must also be a SAS variable in the data set. Each observation in a DATA= data set must contain a measurement for each *process* and a value for the *subgroup-variable*. Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the DATA= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option in the IRCHART statement (for an example, see "Displaying Stratification in Phases" on page 1936).

For an example of a DATA= data set, see "Creating Individual Measurements and Moving Range Charts" on page 1348.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set CONLIMS:*

*In Release 6.09 and in earlier releases, it is necessary to specify the READLIMITS option.

```
proc shewhart data=info limits=conlims;
   irchart weight*id;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set; see Table 41.23 on page 1372. The LIMITS= data set can also be created directly using a DATA step.

When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLI_, _MEAN_, _UCLI_, _LCLR_, _R_, and _UCLR_, which specify the control limits directly
- the variables _MEAN_ and _STDDEV_, which are used to calculate the control limits according to the equations in Table 41.23 on page 1372

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are `ESTIMATE`, `STANDARD`, `STDMU`, and `STDSIGMA`. See Example 41.2 on page 1383 for an illustration.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1354.

HISTORY= Data Set

You can read individual measurements and moving ranges from a HISTORY= data set specified in the PROC SHEWHART statement. This allows you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART procedure.

A HISTORY= data set used with the IRCHART statement must contain the following:

- the *subgroup-variable*
- an individual measurements variable for each *process*
- a moving range variable for each *process*

The name of the individual measurements variable must be the *process* specified in the IRCHART statement. The name of the moving range variable must be the prefix *process* concatenated with the special suffix character R. For example, consider the following statements:

```
proc shewhart history=summary;
   irchart (weight yldstren)*id;
run;
```

The data set SUMMARY must include the variables ID, WEIGHT, WEIGHTR, YLDSTREN, and YLDSRENR.

Note that if you specify a *process* name that contains 32 characters, the name of the moving range variable must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with R.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 1936 for an example).

For an example of a HISTORY= data set, see "Reading Individual Measurements and Moving Ranges" on page 1351.

TABLE= Data Set

You can read individual measurements, moving ranges, and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in Chapter 56, "Specialized Control Charts."

The following table lists the variables required in a TABLE= data set used with the IRCHART statement:

Table 41.26. Variables Required in a TABLE= Data Set

Variable	Description
LCLI	lower control limit for individual measurements
LCLR	lower control limit for moving range
LIMITN	number of consecutive measurements used to calculate moving ranges
MEAN	process mean
_R_	average moving range
subgroup-variable	values of the *subgroup-variable*
SUBI	individual measurements
SUBR	moving ranges
UCLI	upper control limit for individual measurements
UCLR	upper control limit for moving range

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1352.

Methods for Estimating the Standard Deviation

When control limits are computed from the input data, three methods (referred to as default, MAD and MMR) are available for estimating the process standard deviation σ.

Default Method

The default estimate for σ is

$$\hat{\sigma} = \bar{R}/d_2(n)$$

where $\bar{R}$ is the average of the moving ranges, n is the number of consecutive individual measurements used to compute each moving range, and the unbiasing factor $d_2(n)$ is defined so that if the observations are normally distributed, the expected value of R_i is

$$E(R_i) = d_2(n_i)\sigma$$

This method is described in the *ASTM Manual on Presentation of Data and Control Chart Analysis* (1976).

MAD Method

If you specify SMETHOD=MAD, a median absolute deviation estimator is computed for σ, as described by Boyles (1997). It is computed as

$$\hat{\sigma} = \text{median}\{|X_i - \tilde{X}|, 1 \leq i \leq N\}/0.6745$$

where $\tilde{X}$ is the sample median.

MMR Method

If you specify SMETHOD=MMR, a median moving range estimator is computed for σ. This estimator is described by Boyles (1997). It is computed as

$$\hat{\sigma} = \tilde{R}/0.954$$

where $\tilde{R}$ is the median of the nonmissing moving ranges.

Interpreting Charts for Individual Measurements and Moving Ranges

Montgomery (1996) points out that a moving range chart should be interpreted with care because "the moving ranges are correlated, and this correlation may often induce a pattern or runs or cycles on the chart." For this reason Nelson (1982) recommends against plotting the moving ranges. Nelson notes that the assumption of normality is more critical for an individual measurements chart than for an $\bar{X}$ chart. You can use the NOCHART2 option in the IRCHART statement to specify that only the individual measurements chart is to be displayed. See Example 41.3 on page 1386 for an illustration. If, instead, you specify the SEPARATE option, the charts for individual measurements and moving ranges are displayed on separate screens.

An alternative method for creating an individual measurements chart is to use the XCHART statement, which uses an estimate of σ based on moving ranges of two consecutive measurements when the subgroup sample sizes are all equal to one. Note that the XCHART statement displays the control limit legend $n = 1$ to indicate the common subgroup sample size, whereas the IRCHART statement displays a legend that indicates the number of consecutive measurements used to compute the moving ranges (the "pseudo subgroup sample size").

Nelson (1982) explains that the reason for estimating the process standard deviation σ from moving ranges of two consecutive measurements rather than the sample standard deviation of the measurements is that "the moving range of two minimizes inflationary effects on the variability which are caused by trends and oscillations that may be present." Nelson suggests that any moving range that exceeds 3.5 times the average moving range should be removed from the calculation of the average moving range.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical (Individual measurements chart)	DATA=	*process*
Vertical (Individual measurements chart)	HISTORY=	subgroup measurement variable
Vertical (Individual measurements chart)	TABLE=	_SUBI_

You can specify distinct labels for the vertical axes of the individual measurements and moving range charts by breaking the vertical axis into two parts with a split character. Specify the split character with the SPLIT= option. The first part labels the vertical axis of the individual measurements chart, and the second part labels the vertical axis of the moving range chart.

For example, the following sets of statements specify the label *Avg gap in mm* for the vertical axis of the individual measurements chart and the label *Range in mm* for the vertical axis of the moving range chart:

```
proc shewhart data=doors;
   irchart gap*hour / split = '/' ;
   label gap = 'Avg gap in mm/Range in mm';
run;

proc shewhart history=doorhist;
   irchart gap*hour / split = '/' ;
   label gap = 'Avg gap in mm/Range in mm';
run;

proc shewhart table=doortab;
   irchart gap*hour / split = '/' ;
   label _SUBI_ = 'Avg gap in mm/Range in mm';
run;
```

In this example, the label assignments are in effect only for the duration of the procedure step, and they temporarily override any permanent labels associated with the variables.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples

This section provides advanced examples of the IRCHART statement.

Example 41.1. Applying Tests for Special Causes

See SHWIREX1 in the SAS/QC Sample Library

This example illustrates how you can apply tests for special causes to make an individual measurements chart more sensitive to special causes of variation. The following statements create a data set named ENGINES, which contains the weights for 25 jet engines:

```
data engines;
   input id weight @@;
   label weight='Engine Weight (lbs)'
         id    ='Engine ID Number';
   datalines;
1711 1270   1712 1258   1713 1248   1714 1260
1715 1263   1716 1260   1717 1259   1718 1240
1719 1260   1720 1246   1721 1238   1722 1253
1723 1249   1724 1245   1725 1251   1726 1252
1727 1249   1728 1274   1729 1258   1730 1268
1731 1248   1732 1295   1733 1243   1734 1253
1735 1258
;
run;
```

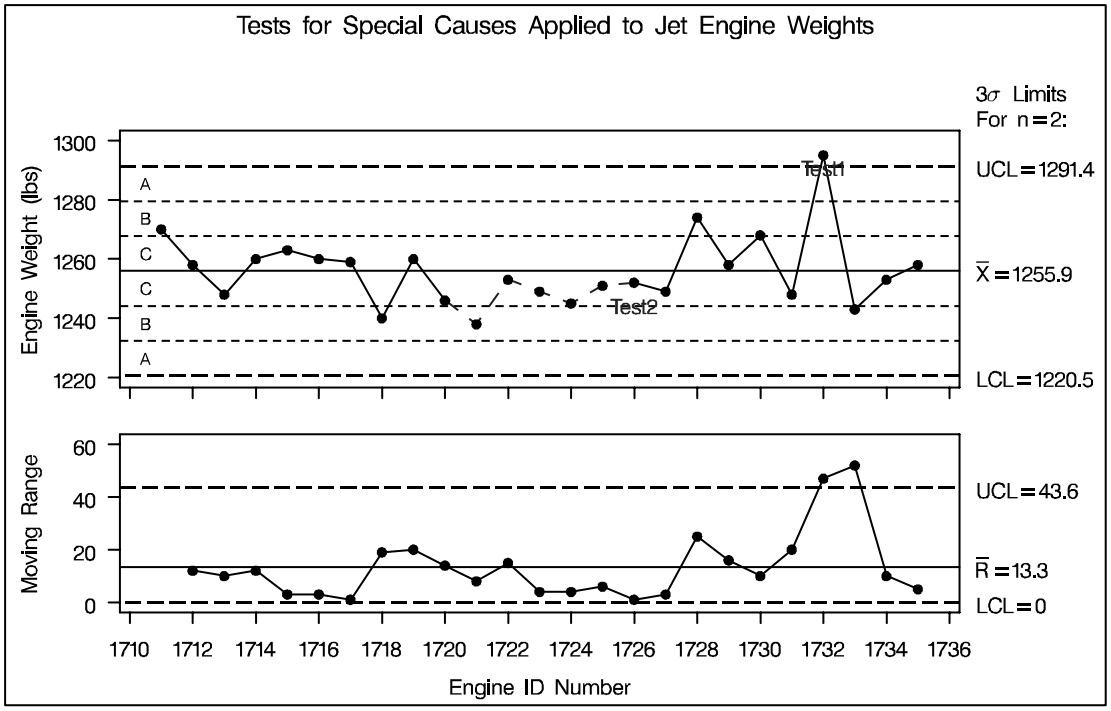
Individual measurements and moving range charts are used to monitor the weights. The following statements produce the tables shown in Output 41.1.1 and create the charts shown in Output 41.1.2:

```
title 'Tests for Special Causes Applied to Jet Engine Weights';
symbol v=dot;
proc shewhart data=engines;
   irchart weight*id /
       tests   =1 to 8
       test2run=7
       tabletests
       zonelabels
       ltests  =20;
run;
```

The TESTS= option applies eight tests for special causes, which are described in Chapter 55, "Tests for Special Causes." The TEST2RUN= option specifies the length of the pattern for Test 2. The TABLETESTS option requests a table of individual measurements, moving ranges, and control limits, and it adds a column indicating which measurements tested positive for special causes.

The ZONELABELS option displays zone lines and zone labels on the individual measurements chart. The zones are used to define the tests. The LTESTS= option specifies the line type used to connect the points in a pattern for a test that is signaled.

Output 41.1.1. Tabular Form of Individual Measurements and Moving Range Chart



Output 41.1.1 and Output 41.1.2 indicate that Test 1 was positive for engine 1732 and Test 2 was positive for engine 1726. Test 1 detects one point beyond Zone A (outside the control limits) and Test 2 detects seven points (TEST2RUN=7) in a row on one side of the central line.

Example 41.2. Specifying Standard Values for the Process Mean and Standard Deviation

See SHWIREX2 in the SAS/QC Sample Library

By default, the IRCHART statement estimates the process mean (μ) and standard deviation (σ) from the data, as in the previous example. However, there are applications in which known (standard) values μ_0 and σ_0 are available for these parameters based on previous experience or extensive sampling.

For example, suppose that the manufacturing process described in the previous example produces engines whose weights are normally distributed with a mean of 1250 and a standard deviation of 12. The following statements create individual measurements and moving range charts based on these values:

```
symbol v = dot h = .8;
title 'Specifying Standard Process Mean and Standard Deviation';
proc shewhart data=engines;
   irchart weight*id /
       mu0      = 1250
       sigma0   = 12
       xsymbol  = mu0;
run;
```

Output 41.1.2. Tests for Special Causes

Tests for Special Causes Applied to Jet Engine Weights

Individual Measurements Chart Summary for weight

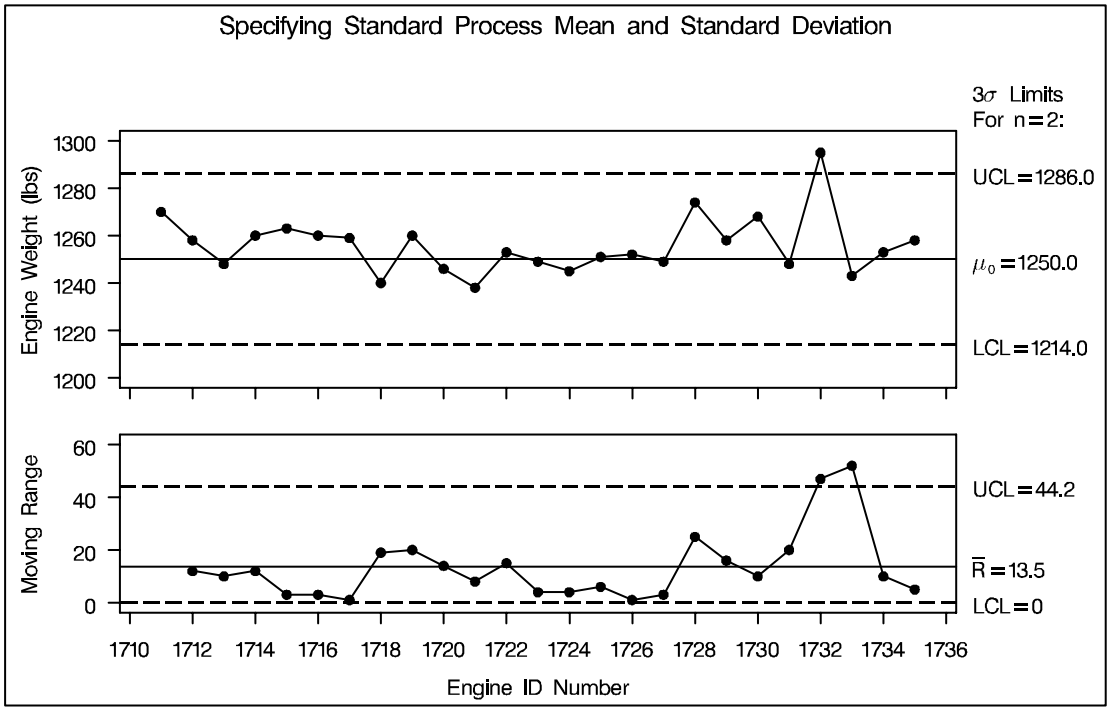
id	--3 Sigma Limits with n=2 for weight-- Lower Limit	weight	Upper Limit	Special Tests Signaled
1711	1220.4709	1270.0000	1291.3691	
1712	1220.4709	1258.0000	1291.3691	
1713	1220.4709	1248.0000	1291.3691	
1714	1220.4709	1260.0000	1291.3691	
1715	1220.4709	1263.0000	1291.3691	
1716	1220.4709	1260.0000	1291.3691	
1717	1220.4709	1259.0000	1291.3691	
1718	1220.4709	1240.0000	1291.3691	
1719	1220.4709	1260.0000	1291.3691	
1720	1220.4709	1246.0000	1291.3691	
1721	1220.4709	1238.0000	1291.3691	
1722	1220.4709	1253.0000	1291.3691	
1723	1220.4709	1249.0000	1291.3691	
1724	1220.4709	1245.0000	1291.3691	
1725	1220.4709	1251.0000	1291.3691	
1726	1220.4709	1252.0000	1291.3691	2
1727	1220.4709	1249.0000	1291.3691	
1728	1220.4709	1274.0000	1291.3691	
1729	1220.4709	1258.0000	1291.3691	
1730	1220.4709	1268.0000	1291.3691	
1731	1220.4709	1248.0000	1291.3691	
1732	1220.4709	1295.0000	1291.3691	1
1733	1220.4709	1243.0000	1291.3691	
1734	1220.4709	1253.0000	1291.3691	
1735	1220.4709	1258.0000	1291.3691	

Individual Measurements Chart Summary for weight

id	--3 Sigma Limits with n=2 for Moving Range- Lower Limit	Moving Range	Upper Limit
1711	0	.	43.553759
1712	0	12.000000	43.553759
1713	0	10.000000	43.553759
1714	0	12.000000	43.553759
1715	0	3.000000	43.553759
1716	0	3.000000	43.553759
1717	0	1.000000	43.553759
1718	0	19.000000	43.553759
1719	0	20.000000	43.553759
1720	0	14.000000	43.553759
1721	0	8.000000	43.553759
1722	0	15.000000	43.553759
1723	0	4.000000	43.553759
1724	0	4.000000	43.553759
1725	0	6.000000	43.553759
1726	0	1.000000	43.553759
1727	0	3.000000	43.553759
1728	0	25.000000	43.553759
1729	0	16.000000	43.553759
1730	0	10.000000	43.553759
1731	0	20.000000	43.553759
1732	0	47.000000	43.553759
1733	0	52.000000	43.553759
1734	0	10.000000	43.553759
1735	0	5.000000	43.553759

The charts are shown in Output 41.2.1. The MU0= option and SIGMA0= option specify μ_0 and σ_0. The XSYMBOL= option specifies the label for the central line on the individual measurements chart, and the keyword MU0 requests a label indicating that the central line is based on a standard value.

Output 41.2.1. Specifying Standard Values with MU0= and SIGMA0=



You can also specify μ_0 and σ_0 as the values of the variables _MEAN_ and _STDDEV_ in a LIMITS= data set. For example, the following statements create a LIMITS= data set with the standard values specified in the preceding IRCHART statement:

```
data englim;
   length _var_ _subgrp_ _type_ $8;
   _var_    = 'weight';
   _subgrp_ = 'id';
   _limitn_ = 2;
   _type_   = 'STANDARD';
   _mean_   = 1250;
   _stddev_ = 12;
run;
```

The variables _VAR_ and _SUBGRP_ are required, and their values must match the *process* and *subgroup-variable*, respectively, specified in the IRCHART statement. The bookkeeping variable _TYPE_ is not required, but it is recommended to indicate that the variables _MEAN_ and _STDDEV_ provide standard values rather than estimated values. See "LIMITS= Data Set" on page 1376 for details.

The following statements read ENGLIM as a LIMITS= data set:

```
proc shewhart data=engines limits=englim;
   irchart weight*id / xsymbol=mu0;
run;
```

The resulting charts (not shown here) are identical to those shown in Output 41.2.1.

Example 41.3. Displaying Distributional Plots in the Margin

See SHWIREX3 in the SAS/QC Sample Library

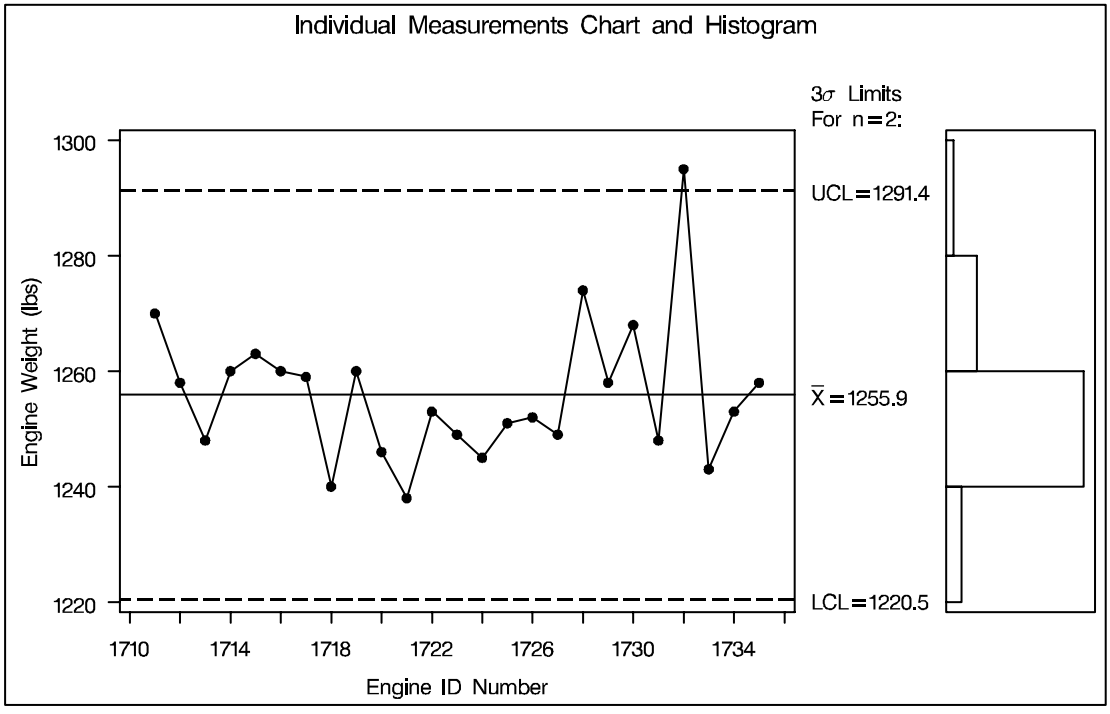
You can augment a chart for individual measurements with one of several graphical displays, such as a histogram or a box-and-whisker plot. These displays summarize the measurements plotted on the chart, and, if the process is in statistical control, they provide a view of the process distribution.

For example, the following statements create an individual measurements chart for the engine weight measurements in the data set ENGINES (see page 1382) augmented with a histogram of the weights:

```
symbol v = dot;
title 'Individual Measurements Chart and Histogram';
proc shewhart data=engines;
   irchart weight*id /
       rtmplot  = histogram
       nochart2;
run;
```

The chart is shown in Output 41.3.1. The RTMPLOT= option requests a histogram in the right margin. The NOCHART2 option suppresses the display of the moving range chart.

Output 41.3.1. Histogram in Right Margin



The following *keywords*, requesting different types of plots, are available with the RTMPLOT= option:

Keyword	Marginal Plot
HISTOGRAM	histogram
DIGIDOT	digidot plot
SKELETAL	skeletal box-and-whisker plot
SCHEMATIC	schematic box-and-whisker plot
SCHEMATICID	schematic box-and-whisker plot with outliers labeled
SCHEMATICIDFAR	schematic box-and-whisker plot with far outliers labeled

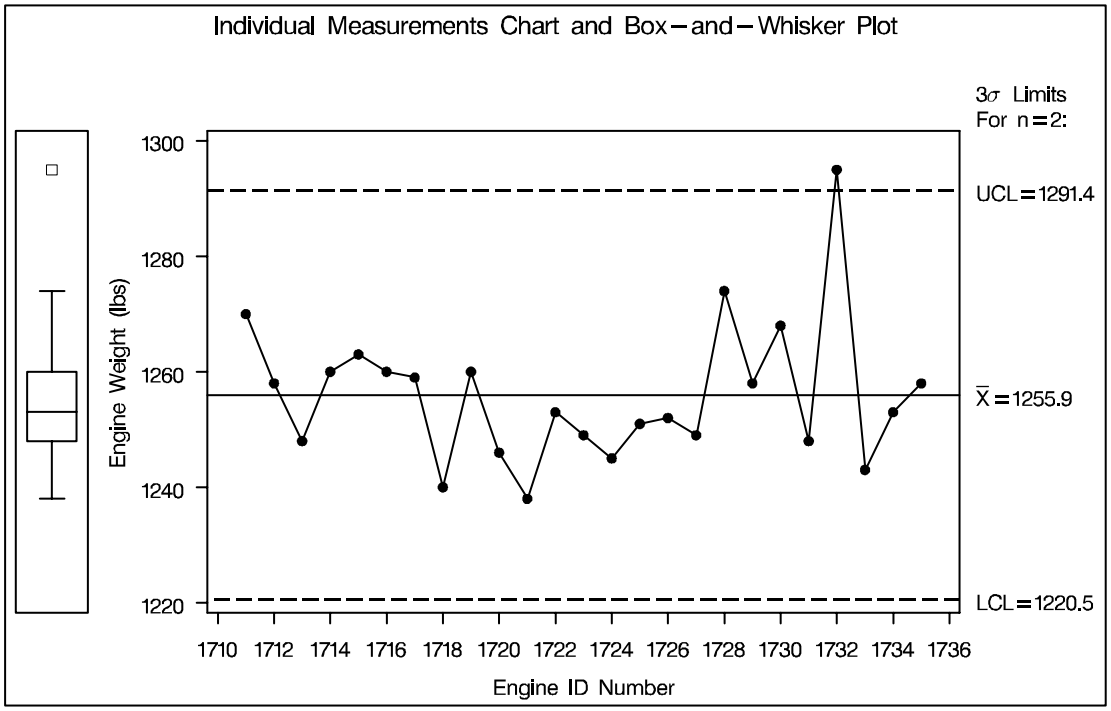
See the entry for the BOXSTYLE= option in Chapter 53, "Dictionary of Options," for a description of the various box-and-whisker plots.

You can also use the LTMPLOT= option to request univariate plots in the left margin. The following statements request an individual measurements chart with a box-and-whisker plot in the left margin:

```
title 'Individual Measurements Chart and Box-and-Whisker Plot';
proc shewhart data=engines;
   irchart weight*id /
       ltmplot  = schematic
       ltmargin = 8
       nochart2;
run;
```

The chart is shown in Output 41.3.2. The same *keywords* that are available with the RTMPLOT= option can be specified with the LTMPLOT= option. The LTMARGIN= option specifies the width (in horizontal percent screen units) of the left margin.

Output 41.3.2. Box-and-Whisker Plot in Left Margin



Chapter 42
MCHART Statement

Chapter Contents

OVERVIEW 1391

GETTING STARTED 1392
- Creating Charts for Medians from Raw Data 1392
- Creating Charts for Medians from Subgroup Summary Data 1394
- Saving Summary Statistics 1398
- Saving Control Limits 1400
- Reading Preestablished Control Limits 1403

SYNTAX 1405
- Summary of Options 1406

DETAILS 1417
- Constructing Median Charts 1417
- Output Data Sets 1419
- ODS Tables 1422
- Input Data Sets 1422
- Methods for Estimating the Standard Deviation 1426
- Axis Labels 1426
- Missing Values 1426

EXAMPLES 1427
- Example 42.1. Controlling Value of Central Line 1427
- Example 42.2. Estimating the Process Standard Deviation 1431

Chapter 42
MCHART Statement

Overview

The MCHART statement creates a chart for subgroup medians, which is used to monitor the central tendency of a process.

You can use options in the MCHART statement to

- compute control limits from the data based on a multiple of the standard error of the plotted medians or as probability limits
- tabulate subgroup sample sizes, subgroup medians, control limits, and other information
- save control limits in an output data set
- save subgroup sample sizes and subgroup medians in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify one of several methods for estimating the process standard deviation
- specify whether subgroup standard deviations or subgroup ranges are used to estimate the process standard deviation
- specify a known (standard) process mean and standard deviation for computing control limits
- create a secondary chart that displays a time trend removed from the data (see "Displaying Trends in Process Data" on page 1957)
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the charts more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

Note: When analyzing variables data, you should examine the variability of the process as well as the mean level. You can use the MRCHART statement in the SHEWHART procedure to monitor both the mean level and variability.

Getting Started

This section introduces the MCHART statement with simple examples that illustrate commonly used options. Complete syntax for the MCHART statement is presented in the "Syntax" section on page 1405.

Creating Charts for Medians from Raw Data

See SHWMCHR in the SAS/QC Sample Library

A consumer products company weighs detergent boxes (in pounds) to determine whether the fill process is in control. The following statements create a SAS data set named DETERGENT, which contains the weights for five boxes in each of 28 lots. A lot is considered a rational subgroup.

```
data detergent;
   input lot @;
   do i=1 to 5;
      input weight @;
      output;
   end;
drop i;
datalines;
 1 17.39 26.93 19.34 22.56 24.49
 2 23.63 23.57 23.54 20.56 22.17
 3 24.35 24.58 23.79 26.20 21.55
 4 25.52 28.02 28.44 25.07 23.39
 5 23.25 21.76 29.80 23.09 23.70
 6 23.01 22.67 24.70 20.02 26.35
 7 23.86 24.19 24.61 26.05 24.18
 8 26.00 26.82 28.03 26.27 25.85
 9 21.58 22.31 25.03 20.86 26.94
10 22.64 21.05 22.66 29.26 25.02
11 26.38 27.50 23.91 26.80 22.53
12 23.01 23.71 25.26 20.21 22.38
13 23.15 23.53 22.98 21.62 26.99
14 26.83 23.14 24.73 24.57 28.09
15 26.15 26.13 20.57 25.86 24.70
16 25.81 23.22 23.99 23.91 27.57
17 25.53 22.87 25.22 24.30 20.29
18 24.88 24.15 25.29 29.02 24.46
19 22.32 25.96 29.54 25.92 23.44
20 25.63 26.83 20.95 24.80 27.25
21 21.68 21.11 26.07 25.17 27.63
22 26.72 27.05 24.90 30.08 25.22
23 31.58 22.41 23.67 23.47 24.90
24 28.06 23.44 24.92 24.64 27.42
25 21.10 22.34 24.96 26.50 24.51
26 23.80 24.03 24.75 24.82 27.21
27 25.10 26.09 27.21 24.28 22.45
28 25.53 22.79 26.26 25.85 25.64
;
run;
```

A partial listing of DETERGENT is shown in Figure 42.1.

The Data Set DETERGENT

lot	weight
1	17.39
1	26.93
1	19.34
1	22.56
1	24.49
2	23.63
2	23.57
2	23.54
2	20.56
2	22.17
3	24.35
3	24.58
3	23.79
3	26.20
3	21.55
.	.
.	.
.	.

Figure 42.1. Partial Listing of the Data Set DETERGENT

The data set DETERGENT is said to be in "strung-out" form, since each observation contains the lot number and weight of a single box. The first five observations contain the weights for the first lot, the second five observations contain the weights for the second lot, and so on. Because the variable LOT classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable WEIGHT contains the weights and is referred to as the *process variable* (or *process* for short).

The within-subgroup variability of the weights is known to be stable. You can use a median chart to determine whether the mean level of the weights is in control. The following statements create the median chart shown in Figure 42.2:

```
title 'Median Chart for Detergent Box Weight';
proc shewhart data=detergent;
   mchart weight*lot;
run;
```

This example illustrates the basic form of the MCHART statement. After the keyword MCHART, you specify the *process* to analyze (in this case, WEIGHT) followed by an asterisk and the *subgroup-variable* (LOT).

The input data set is specified with the DATA= option in the PROC SHEWHART statement.

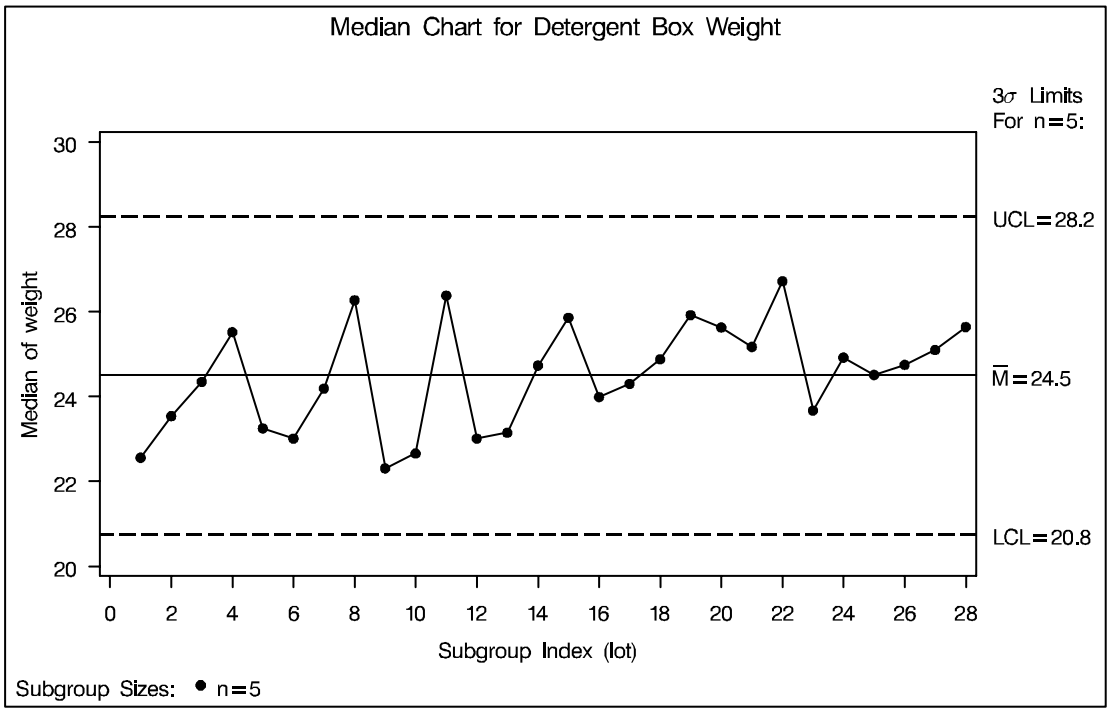


Figure 42.2. Median Chart for Detergent Box Weight Data

Each point on the chart represents the median of the weights for a particular lot. For instance, the weights for the first lot are 17.39, 19.34, 22.56, 24.49, and 26.93, and consequently, the median plotted for this lot is 22.56.

Since all of the subgroup medians lie within the control limits, you can conclude that the process is in statistical control. By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 42.23 on page 1418. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1403.

For computational details, see "Constructing Median Charts" on page 1417. For more details on reading raw measurements, see "DATA= Data Set" on page 1422.

Creating Charts for Medians from Subgroup Summary Data

See SHWMCHR in the SAS/QC Sample Library

The previous example illustrates how you can create median charts using raw data (process measurements). However, in many applications the data are provided as subgroup summary statistics. This example illustrates how you can use the MCHART statement with data of this type.

The following data set (DETSUM) provides the data from the preceding example in summarized form. There is exactly one observation for each subgroup (note that the subgroups are still indexed by LOT). The variable WEIGHTM contains the subgroup medians, the variable WEIGHTR contains the subgroup ranges, and the variable WEIGHTN contains the subgroup sample sizes (these are all five).

```
data detsum;
   input lot weightm weightr;
   weightn = 5;
datalines;
 1  22.56  9.54
 2  23.54  3.07
 3  24.35  4.65
 4  25.52  5.05
 5  23.25  8.04
 6  23.01  6.33
 7  24.19  2.19
 8  26.27  2.18
 9  22.31  6.08
10  22.66  8.21
11  26.38  4.97
12  23.01  5.05
13  23.15  5.37
14  24.73  4.95
15  25.86  5.58
16  23.99  4.35
17  24.30  5.24
18  24.88  4.87
19  25.92  7.22
20  25.63  6.30
21  25.17  6.52
22  26.72  5.18
23  23.67  9.17
24  24.92  4.62
25  24.51  5.40
26  24.75  3.41
27  25.10  4.76
28  25.64  3.47
;
run;
```

A partial listing of DETSUM is shown in Figure 42.3.

Summary Data Set for Detergent Box Weights

lot	weightm	weightr	weightn
1	22.56	9.54	5
2	23.54	3.07	5
3	24.35	4.65	5
4	25.52	5.05	5
5	23.25	8.04	5
6	23.01	6.33	5
7	24.19	2.19	5
8	26.27	2.18	5
9	22.31	6.08	5
10	22.66	8.21	5
11	26.38	4.97	5
12	23.01	5.05	5
13	23.15	5.37	5
14	24.73	4.95	5
15	25.86	5.58	5
16	23.99	4.35	5
17	24.30	5.24	5
18	24.88	4.87	5
19	25.92	7.22	5
20	25.63	6.30	5
21	25.17	6.52	5
22	26.72	5.18	5
23	23.67	9.17	5
24	24.92	4.62	5
25	24.51	5.40	5
26	24.75	3.41	5
27	25.10	4.76	5
28	25.64	3.47	5

Figure 42.3. The Summary Data Set DETSUM

You can read this data set by specifying it as a HISTORY= data set in the PROC SHEWHART statement, as follows:

```
title 'Median Chart for Weights';
proc shewhart history=detsum lineprinter;
   mchart weight*lot='*';
run;
```

The resulting median chart is shown in Figure 42.4. Since the LINEPRINTER option is included in the PROC SHEWHART statement, line printer output is produced. The asterisk (*) specified in single quotes after the *subgroup-variable* indicates the character used to plot points. This character must follow an equal sign.

Note that WEIGHT is *not* the name of a SAS variable in the data set DETSUM but is, instead, the common prefix for the names of the three SAS variables WEIGHTM, WEIGHTR, and WEIGHTN. The suffix characters *M*, *R*, and *N* indicate *median*, *range*, and *sample size*, respectively. Thus, you can specify three subgroup summary variables in the HISTORY= data set with a single name (WEIGHT), which is referred to as the *process*. The name LOT specified after the asterisk is the name of the *subgroup-variable*.

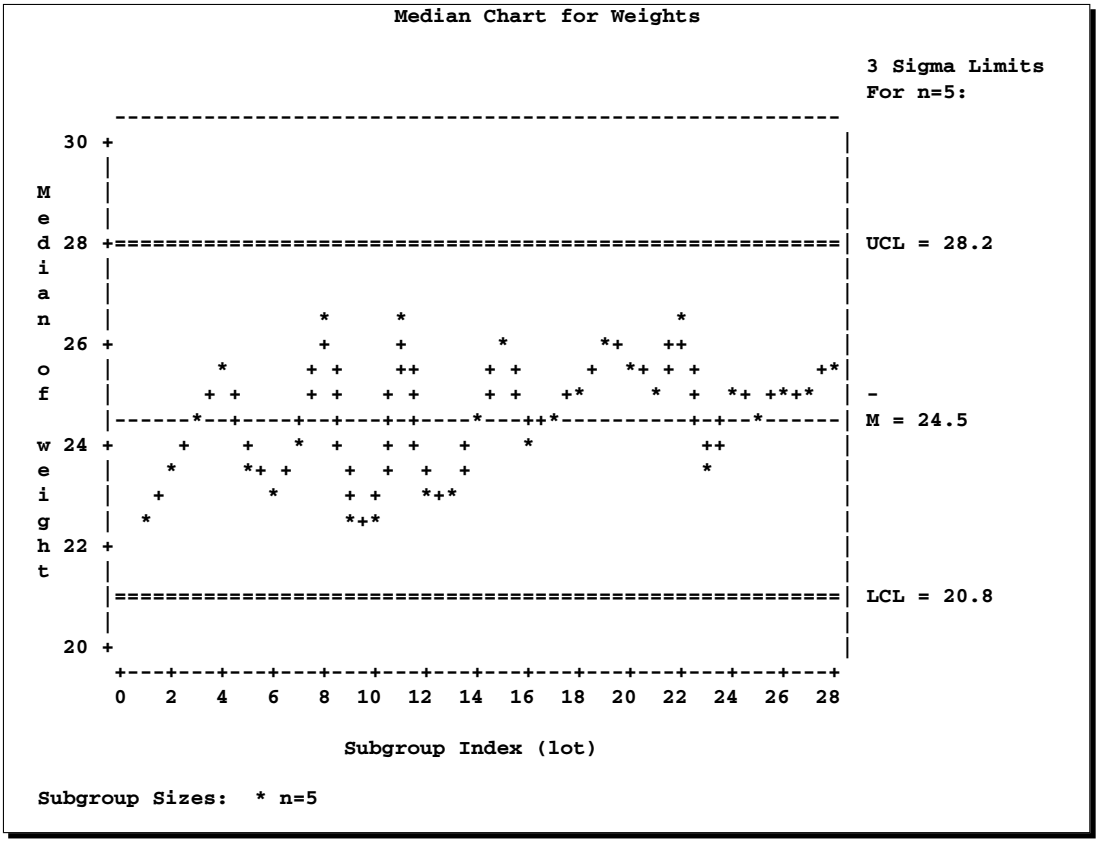


Figure 42.4. Median Chart from the Summary Data Set DETSUM

In general, a HISTORY= input data set used with the MCHART statement must contain the following variables:

- subgroup variable
- subgroup median variable
- either a subgroup range variable or a subgroup standard deviation variable
- subgroup sample size variable

Furthermore, the names of the subgroup median, range (or standard deviation), and sample size variables must begin with the *process* name specified in the MCHART statement and end with the special suffix characters *M*, *R* (or *S*), and *N*, respectively. If the names do not follow this convention, you can use the RENAME option in the PROC SHEWHART statement to rename the variables for the duration of the SHEWHART procedure step (see page 1442).

If you specify the STDDEVIATIONS option in the MCHART statement, the HISTORY= data set must contain a subgroup standard deviation variable; otherwise, the HISTORY= data set must contain a subgroup range variable. The STDDEVIATIONS option specifies that the estimate of the process standard deviation σ is to be calculated from subgroup standard deviations rather than subgroup

ranges. For example, in the following statements, the data set DETSUM2 must contain a subgroup standard deviation variable named WEIGHTS:

```
title 'Median Chart for Weights';
symbol v=dot;
proc shewhart history=detsum2;
   mchart weight*lot / stddeviations;
run;
```

Options such as STDDEVIATIONS are specified after the slash (/) in the MCHART statement. A complete list of options is presented in the "Syntax" section on page 1405.

In summary, the interpretation of *process* depends on the input data set.

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names of the variables containing the summary statistics.

For more information, see "HISTORY= Data Set" on page 1424.

Saving Summary Statistics

See SHWMCHR in the SAS/QC Sample Library

In this example, the MCHART statement is used to create a summary data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read measurements from the data set DETERGENT and create a summary data set named DETHIST:

```
proc shewhart data=detergent;
   mchart weight*lot / outhistory = dethist
                       nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 42.2. Figure 42.5 contains a partial listing of DETHIST.

Summary Data Set DETHIST for Detergent Box Weights

lot	weight M	weight R	weight N
1	22.56	9.54	5
2	23.54	3.07	5
3	24.35	4.65	5
4	25.52	5.05	5
5	23.25	8.04	5
6	23.01	6.33	5
7	24.19	2.19	5
8	26.27	2.18	5
9	22.31	6.08	5
10	22.66	8.21	5
11	26.38	4.97	5
12	23.01	5.05	5
13	23.15	5.37	5
14	24.73	4.95	5
15	25.86	5.58	5
16	23.99	4.35	5
17	24.30	5.24	5
18	24.88	4.87	5
19	25.92	7.22	5
20	25.63	6.30	5
21	25.17	6.52	5
22	26.72	5.18	5
23	23.67	9.17	5
24	24.92	4.62	5
25	24.51	5.40	5
26	24.75	3.41	5
27	25.10	4.76	5
28	25.64	3.47	5

Figure 42.5. The Summary Data Set DETHIST

There are four variables in the data set DETHIST.

- LOT contains the subgroup index.
- WEIGHTM contains the subgroup medians.
- WEIGHTR contains the subgroup ranges.
- WEIGHTN contains the subgroup sample sizes.

Note that the summary statistic variables are named by adding the suffix characters *M*, *R*, and *N* to the *process* WEIGHT specified in the MCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

If you specify the STDDEVIATIONS option, the OUTHISTORY= data set includes a subgroup standard deviation variable instead of a subgroup range variable, as demonstrated by the following statements:

```
proc shewhart data=detergent;
   mchart weight*lot / outhistory = dethist2
                       stddeviations
                       nochart;
run;
```

Figure 42.6 contains a partial listing of DETHIST2.

Summary Data Set with Subgroup Standard Deviations

lot	weight M	weightS	weight N
1	22.56	3.84205	5
2	23.54	1.34050	5
3	24.35	1.68087	5
4	25.52	2.11558	5
5	23.25	3.14747	5
6	23.01	2.37115	5
7	24.19	0.86491	5
8	26.27	0.88382	5
9	22.31	2.55563	5
10	22.66	3.20064	5
11	26.38	2.10858	5
12	23.01	1.85360	5
13	23.15	1.99936	5
14	24.73	1.96853	5
15	25.86	2.37425	5
16	23.99	1.77395	5
17	24.30	2.14006	5
18	24.88	1.98148	5
19	25.92	2.78591	5
20	25.63	2.51040	5
21	25.17	2.82905	5
22	26.72	2.05752	5
23	23.67	3.67124	5
24	24.92	1.96007	5
25	24.51	2.15219	5
26	24.75	1.35365	5
27	25.10	1.80968	5
28	25.64	1.38345	5

Figure 42.6. The Summary Data Set DETHIST2

The variable WEIGHTS, which contains the subgroup standard deviations, is named by adding the suffix character *S* to the *process* WEIGHT.

For more information, see "OUTHISTORY= Data Set" on page 1420.

Saving Control Limits

See SHWMCHR in the SAS/QC Sample Library

You can save the control limits for a median chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1403) or modify the limits with a DATA step program.

The following statements read measurements from the data set DETERGENT (see page 1392) and save the control limits displayed in Figure 42.2 in a data set named DETLIM:

```
proc shewhart data=detergent;
   mchart weight*lot / outlimits=detlim
                       nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the charts. The data set DETLIM is listed in Figure 42.7.

Control Limits for Detergent Box Weights

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLM_
weight	lot	ESTIMATE	5	.002909021	3	20.7554

MEAN	_UCLM_	_LCLR_	_R_	_UCLR_	_STDDEV_
24.4996	28.2439	0	5.42036	11.4613	2.33041

Figure 42.7. The Data Set DETLIM Containing Control Limit Information

The data set DETLIM contains one observation with the limits for the *process* WEIGHT. The variables _LCLM_ and _UCLM_ contain the lower and upper control limits for the medians, and the variable _MEAN_ contains the central line. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values.

The variables _LCLR_, _R_, and _UCLR_ are not used to create median charts, but they are included so the data set DETLIM can be used to create an R chart; see Chapter 43, "MRCHART Statement," and Chapter 46, "RCHART Statement." If you specify the STDDEVIATIONS option in the MCHART statement, the variables _LCLS_, _S_, and _UCLS_ are included in the OUTLIMITS= data set. These variables can be used to create an s chart; see Chapter 47, "SCHART Statement." For more information, see "OUTLIMITS= Data Set" on page 1419.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=detergent;
   mchart weight*lot / outtable=dtable
                       nochart;
run;
```

The data set DTABLE is listed in Figure 42.8.

Summary Statistics and Control Limit Information

VAR	lot	_SIGMAS_	_LIMITN_	_SUBN_	_LCLM_	_SUBMED_	_MEAN_	_UCLM_	_EXLIM_
weight	1	3	5	5	20.7554	22.56	24.4996	28.2439	
weight	2	3	5	5	20.7554	23.54	24.4996	28.2439	
weight	3	3	5	5	20.7554	24.35	24.4996	28.2439	
weight	4	3	5	5	20.7554	25.52	24.4996	28.2439	
weight	5	3	5	5	20.7554	23.25	24.4996	28.2439	
weight	6	3	5	5	20.7554	23.01	24.4996	28.2439	
weight	7	3	5	5	20.7554	24.19	24.4996	28.2439	
weight	8	3	5	5	20.7554	26.27	24.4996	28.2439	
weight	9	3	5	5	20.7554	22.31	24.4996	28.2439	
weight	10	3	5	5	20.7554	22.66	24.4996	28.2439	
weight	11	3	5	5	20.7554	26.38	24.4996	28.2439	
weight	12	3	5	5	20.7554	23.01	24.4996	28.2439	
weight	13	3	5	5	20.7554	23.15	24.4996	28.2439	
weight	14	3	5	5	20.7554	24.73	24.4996	28.2439	
weight	15	3	5	5	20.7554	25.86	24.4996	28.2439	
weight	16	3	5	5	20.7554	23.99	24.4996	28.2439	
weight	17	3	5	5	20.7554	24.30	24.4996	28.2439	
weight	18	3	5	5	20.7554	24.88	24.4996	28.2439	
weight	19	3	5	5	20.7554	25.92	24.4996	28.2439	
weight	20	3	5	5	20.7554	25.63	24.4996	28.2439	
weight	21	3	5	5	20.7554	25.17	24.4996	28.2439	
weight	22	3	5	5	20.7554	26.72	24.4996	28.2439	
weight	23	3	5	5	20.7554	23.67	24.4996	28.2439	
weight	24	3	5	5	20.7554	24.92	24.4996	28.2439	
weight	25	3	5	5	20.7554	24.51	24.4996	28.2439	
weight	26	3	5	5	20.7554	24.75	24.4996	28.2439	
weight	27	3	5	5	20.7554	25.10	24.4996	28.2439	
weight	28	3	5	5	20.7554	25.64	24.4996	28.2439	

Figure 42.8. The Data Set DTABLE

This data set contains one observation for each subgroup sample. The variables _SUBMED_ and _SUBN_ contain the subgroup medians and subgroup sample sizes. The variables _LCLM_ and _UCLM_ contain the lower and upper control limits, and the variable _MEAN_ contains the central line. The variables _VAR_ and LOT contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1421.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read DTABLE and display a median chart (not shown here) identical to the chart in Figure 42.2 on page 1394:

```
title 'Median Chart for Detergent Box Weight';
proc shewhart table=dtable;
   mchart weight*lot;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see Chapter 56, "Specialized Control Charts,"). For more information, see "TABLE= Data Set" on page 1425.

Reading Preestablished Control Limits

In the previous example, the OUTLIMITS= data set DETLIM saved control limits computed from the measurements in DETERGENT. This example shows how these limits can be applied to new data provided in the following data set:

See SHWMCHR in the SAS/QC Sample Library

```
data detergt2;
   input lot @;
   do i=1 to 5;
      input weight @;
      output;
      end;
   drop i;
   datalines;
29 16.66 27.49 18.87 22.53 24.72
30 23.74 23.67 23.64 20.26 22.09
31 24.56 24.82 23.92 26.67 21.38
32 25.89 28.73 29.21 25.38 23.47
33 23.32 21.61 30.75 23.13 23.82
34 23.04 22.65 24.96 19.64 26.84
35 24.01 24.38 24.86 26.50 24.37
36 26.43 27.36 28.74 26.74 26.27
37 21.41 22.24 25.34 20.59 27.51
38 22.62 20.81 22.64 30.15 25.32
39 26.86 28.14 24.06 27.35 22.49
40 23.03 23.83 25.59 19.85 22.33
41 23.19 23.63 23.00 21.46 27.57
42 27.38 23.18 24.99 24.81 28.82
43 26.60 26.58 20.26 26.27 24.96
44 26.22 23.28 24.15 24.06 28.23
45 25.90 22.88 25.55 24.50 19.95
46 16.66 27.49 18.87 22.53 24.72
47 23.74 23.67 23.64 20.26 22.09
48 24.56 24.82 23.92 26.67 21.38
49 25.89 28.73 29.21 25.38 23.47
50 23.32 21.61 30.75 23.13 23.82
;
run;
```

The following statements create a median chart for the data in DETERGT2 using the control limits in DETLIM:

```
symbol v = dot;
title 'Median Chart for Weights';
proc shewhart data=detergt2 limits=detlim;
   mchart weight*lot;
run;
```

The chart is shown in Figure 42.9. The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default,* this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name WEIGHT
- the value of _SUBGRP_ matches the *subgroup-variable* name LOT

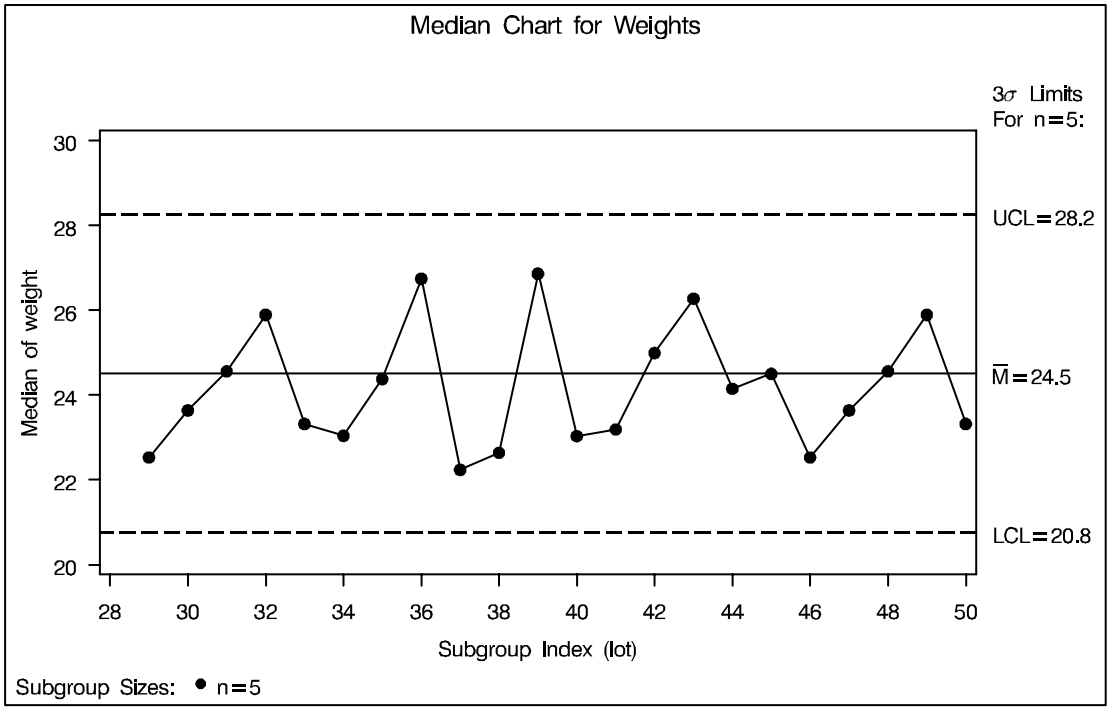


Figure 42.9. Median Chart for Second Set of Detergent Box Weight Data

The chart indicates that the process is in control, since all the medians lie within the control limits.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1423 for details concerning the variables that you must provide.

*In Release 6.09 and in earlier releases, it is also necessary to specify the READLIMITS option to read control limits from a LIMITS= data set.

Syntax

The basic syntax for the MCHART statement is as follows:

MCHART *process***subgroup-variable* **;**

The general form of this syntax is as follows:

MCHART (*processes***)***subgroup-variable* <**(***block-variables* **)** >
< =*symbol-variable* | =*'character'* > < **/** *options* >**;**

You can use any number of MCHART statements in the SHEWHART procedure. The components of the MCHART statement are described as follows.

process
processes
identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating Charts for Medians from Raw Data" on page 1392.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating Charts for Medians from Subgroup Summary Data" on page 1394.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1400.

A *process* is required. If you specify more than one process, enclose the list in parentheses. For example, the following statements request distinct median charts for WEIGHT, LENGTH, and WIDTH:

```
proc shewhart data=measures;
   mchart (weight length width)*day;
run;
```

subgroup-variable
is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding MCHART statement, DAY is the subgroup variable. For details, see "Subgroup Variables" on page 1771.

block-variables
are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 1932 for an example.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the medians.

- If you produce a chart on a line printer, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce a chart on a graphics device, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOLn statements. See "Displaying Stratification in Levels of a Classification Variable" on page 1931 for an example.

character
specifies a plotting character for charts produced on line printers. For example, the following statements create a median chart using an asterisk (*) to plot the points:

```
proc shewhart data=values;
   mchart weight*day='*';
run;
```

options
enhance the appearance of the charts, request additional analyses, save results in data sets, and so on. The "Summary of Options" section, which follows, lists all options by function. Chapter 53, "Dictionary of Options," describes each option in detail.

Summary of Options

The following tables list the MCHART statement options by function. For complete descriptions, see Chapter 53, "Dictionary of Options."

Table 42.1. Tabulation Options

TABLE	creates a basic table of subgroup medians, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUT, and TABLETESTS
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with a column indicating which tests for special causes are positive

Note that specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.

Table 42.2. Options for Specifying Tests for Special Causes

NO3SIGMACHECK	allows tests to be applied with control limits other than 3σ limits
TESTS=*value-list*| *customized-pattern-list*	specifies tests for special causes for the median chart
TEST2RUN=*n*	specifies length of pattern for Test 2
TEST3RUN=*n*	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL='*label*'| (*variable*)|*keyword*	provides labels for points where test is positive
TESTLABEL*n*='*label*'	specifies label for n^{th} test for special causes
TESTNMETHOD= STANDARDIZE	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
TESTRESET=*variable*	allows tests for special causes to be reset for the median chart
ZONELABELS	adds labels A, B, and C to zone lines for median chart
ZONES	adds lines to median chart delineating zones A, B, and C
ZONEVALPOS=*n*	specifies position of ZONEVALUES labels
ZONEVALUES	labels zone lines with their values

Table 42.3. Graphical Options for Displaying Tests for Special Causes

CTESTLABBOX=*color*	specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=*color*| *test-color-list*	specifies color for labels used to identify points where test is positive
CTESTSYMBOL=*color*	specifies color for symbol used to plot points where test is positive
CZONES=*color*	specifies color for lines and labels delineating zones A, B, and C
LTESTS=*linetype*	specifies type of line connecting points where test is positive
LZONES=*linetype*	specifies line type for lines delineating zones A, B, and C
TESTFONT=*font*	specifies software font for labels at points where test is positive
TESTHEIGHT=*value*	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=*symbol*	specifies plot symbol for points where test is positive
TESTSYMBOLHT=*value*	specifies symbol height for points where test is positive
WTESTS=*n*	specifies width of line connecting points where test is positive

Table 42.4. Line Printer Options for Displaying Tests for Special Causes

TESTCHAR='*character*'	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
ZONECHAR='*character*'	specifies character for lines that delineate zones for tests for special causes

Table 42.5. Clipping Options

CCLIP=*color*	specifies color for plot symbol for clipped points
CLIPCHAR='*character*'	specifies plot character for clipped points
CLIPFACTOR=*value*	determines extent to which extreme points are clipped
CLIPLEGEND='*string*'	specifies text for clipping legend
CLIPLEGPOS=*keyword*	specifies position of clipping legend
CLIPSUBCHAR='*character*'	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=*symbol*	specifies plot symbol for clipped points
CLIPSYMBOLHT=*value*	specifies symbol marker height for clipped points

Table 42.6. Specification Limit Options

CIINDICES (ALPHA=*value* TYPE=*keyword*)	specifies α value and type for computing capability index confidence limits
LSL=*value-list*	specifies list of lower specification limits
TARGET=*value-list*	specifies list of target values
USL=*value-list*	specifies list of upper specification limits

Table 42.7. Grid Options

CGRID=*color*	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=*linetype*	specifies line type for grid requested with the ENDGRID option
LGRID=*linetype*	specifies line type for grid requested with the GRID option
WGRID=*n*	specifies width of grid lines

Table 42.8. Reference Line Options

CHREF=*color*	specifies color for lines requested by HREF= and HREF2= options
CVREF=*color*	specifies color for lines requested by VREF= and VREF2= options
HREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on median chart
HREF2=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFCHAR='*character*'	specifies line character for HREF= and HREF2= lines
HREFDATA= *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on median chart
HREF2DATA= *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFLABELS= *'label1'...'labeln'*	specifies labels for HREF= lines
HREF2LABELS= *'label1'...'labeln'*	specifies labels for HREF2= lines
HREFLABPOS=*n*	specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=*linetype*	specifies line type for HREF= and HREF2= lines
LVREF=*linetype*	specifies line type for VREF= and VREF2= lines
NOBYREF	specifies that reference line information in a data set is to be applied uniformly to charts created for all BY groups
VREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to vertical axis on median chart
VREF2=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to vertical axis on trend chart
VREFCHAR='*character*'	specifies line character for VREF= and VREF2= lines
VREFLABELS= *'label1'...'labeln'*	specifies labels for VREF= lines
VREF2LABELS= *'label1'...'labeln'*	specifies labels for VREF2= lines
VREFLABPOS=*n*	specifies position of VREFLABELS= and VREF2LABELS= labels

Table 42.9. Block Variable Legend Options

BLOCKLABELPOS=*keyword*	specifies position of label for *block-variable* legend
BLOCKLABTYPE=*value*|*keyword*	specifies text size of *block-variable* legend
BLOCKPOS=*n*	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=*color-list*	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=*variable*|(*variables*)	specifies one or more variables whose values are colors for filling background of *block-variable* legend

Table 42.10. Axis and Axis Label Options

CAXIS=*color*	specifies color for axis lines and tick marks
CFRAME=*color*|(*color-list*)	specifies fill colors for frame for plot area
CTEXT=*color*	specifies color for tick mark values and axis labels
HAXIS=*values*|AXIS*n*	specifies major tick mark values for horizontal axis
HEIGHT=*value*	specifies height of axis label and axis legend text
HMINOR=*n*	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=*value*	specifies length of offset at both ends of horizontal axis
INTSTART=*value*	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=*n*	specifies thinning factor for tick mark labels on horizontal axis
SPLIT='*character*'	specifies splitting character for axis labels
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=*values*|AXIS*n*	specifies major tick mark values for vertical axis of median chart
VAXIS2=*values*|AXIS*n*	specifies major tick mark values for vertical axis of trend chart
VFORMAT=*format*	specifies format for primary vertical axis tick mark labels
VFORMAT2=*format*	specifies format for secondary vertical axis tick mark labels
VMINOR=*n*	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=*value*	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis for primary chart
VZERO2	forces origin to be included in vertical axis for secondary chart
WAXIS=*n*	specifies width of axis lines

Table 42.11. Options for Specifying Control Limits

ALPHA=*value*	requests probability limits for control charts
LIMITN=*n*|VARYING	specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	computes control limits for each *process* from the data rather than from a LIMITS= data set (Release 6.10 and later releases)
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEXES=ALL| '*label1*'...'*labeln*'	reads multiple sets of control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (Release 6.09 and earlier releases)
SIGMAS=*k*	specifies width of control limits in terms of multiple *k* of standard error of plotted statistic

Table 42.12. Options for Displaying Control Limits

CINFILL=*color*	specifies color for area inside control limits
CLIMITS=*color*	specifies color of control limits, central line, and related labels
LCLLABEL='*label*'	specifies label for lower control limit on a median chart
LIMLABSUBCHAR= '*character*'	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
on median chart	
LLIMITS=*linetype*	specifies line type for control limits
NDECIMAL=*n*	specifies number of digits to right of decimal place in default labels for control limits and central line in median chart
NOCTL	suppresses display of central line in median chart
NOLCL	suppresses display of lower control limit in median chart
NOLIMITLABEL	suppresses labels for control limits and central line
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOUCL	suppresses display of upper control limit in median chart
UCLLABEL='*string*'	specifies label for upper control limit in median chart
WLIMITS=*n*	specifies width for control limits and central line
XSYMBOL='*string*'| *keyword*	specifies label for central line in median chart

Table 42.13. Options for Plotting and Labeling Points

ALLLABEL=VALUE|(*variable*)	labels every point on median chart
ALLLABEL2=VALUE|(*variable*)	labels every point on trend chart
CLABEL=*color*	specifies color for labels
CCONNECT=*color*	specifies color for line segments that connect points on chart
CFRAMELAB=*color*	specifies fill color for frame around labeled points
CNEEDLES=*color*	specifies color for needles that connect points to central line
CONNECTCHAR='*character*'	specifies character used to form line segments that connect points on chart
COUT=*color*	specifies color for portions of line segments that connect points outside control limits
COUTFILL=*color*	specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=*angle*	specifies angle at which labels are drawn
LABELFONT=*font*	specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=*value*	specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on chart
NOTRENDCONNECT	suppresses line segments that connect points on trend chart
OUTLABEL=VALUE|(*variable*)	labels points outside control limits on median chart
SYMBOLCHARS='*characters*'	specifies characters indicating *symbol-variable*
SYMBOLLEGEND=NONE|*name*	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=*keyword*	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=*n*	specifies width of needles

Table 42.14. Process Mean and Standard Deviation Options

MEDCENTRAL=*keyword*	specifies method for estimating process mean μ
MU0=*value*	specifies known value μ_0 for process mean μ
SIGMA0=*value*	specifies known value σ_0 for process standard deviation σ
SMETHOD=*keyword*	specifies method for estimating process standard deviation σ
STDDEVIATIONS	specifies that estimate of process standard deviation σ is to be calculated from subgroup standard deviations
TYPE=*keyword*	identifies whether parameters are estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set

Table 42.15. Phase Options

CPHASELEG=*color*	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE='*string*'	specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=*value*| *keyword*	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES= ALL| '*label1*'...'*labeln*'	specifies *phases* to be read from an input data set

Table 42.16. Input Data Set Options

MISSBREAK	specifies that observations with missing values are not to be processed

Table 42.17. Output Data Set Options

OUTHISTORY= *SAS-data-set*	creates output data set containing subgroup summary statistics
OUTINDEX='*string*'	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS= *SAS-data-set*	creates output data set containing control limits
OUTTABLE= *SAS-data-set*	creates output data set containing subgroup summary statistics and control limits

Table 42.18. Plot Layout Options

ALLN	plots summary statistics for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts for a process variable only when exceptions occur
INTERVAL=*keyword*	specifies natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=*n*	specifies maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of median chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=*n*	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=*n*	specifies number of pages or screens to be used to display chart
TRENDVAR=*variable*|(*variable-list*)	specifies list of trend variables
YPCT1=*value*	specifies length of vertical axis on median chart as a percentage of sum of lengths of vertical axes for median and trend charts
ZEROSTD	displays median chart regardless of whether $\hat{\sigma} = 0$

Table 42.19. Graphical Enhancement Options

ANNOTATE=*SAS-data-set*	specifies annotate data set that adds features to median chart
ANNOTATE2=*SAS-data-set*	specifies annotate data set that adds features to trend chart
DESCRIPTION='*string*'	specifies string that appears in the description field of PROC GREPLAY master menu for median chart
FONT=*font*	specifies software font for labels and legends on charts
NAME='*string*'	specifies name that appears in the name field of the PROC GREPLAY master menu for median chart
PAGENUM='*string*'	specifies the form of the label used in pagination
PAGENUMPOS=*keyword*	specifies the position of the page number requested with the PAGENUM= option
WTREND=*n*	specifies width of line segments connecting points on trend chart

Table 42.20. Star Options

CSTARCIRCLES=*color*	specifies color for STARCIRCLES= circles
CSTARFILL=*color*| *(variable)*	specifies color for filling stars
CSTAROUT=*color*	specifies outline color for stars exceeding inner or outer circles
CSTARS=*color*| *(variable)*	specifies color for outlines of stars
LSTARCIRCLES= *linetypes*	specifies line types for STARCIRCLES= circles
LSTARS=*linetype*|*(variable)*	specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=*value*	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=*value-list*	specifies reference circles for stars
STARINRADIUS=*value*	specifies inner radius of stars
STARLABEL=*keyword*	specifies vertices to be labeled
STARLEGEND=*keyword*	specifies style of legend for star vertices
STARLEGENDLAB=*'label'*	specifies label for STARLEGEND= legend
STAROUTRADIUS=*value*	specifies outer radius of stars
STARSPEC=*value*| *SAS-data-set*	specifies method used to standardize vertex variables
STARSTART=*value*	specifies angle for first vertex
STARTYPE=*keyword*	specifies graphical style of star
STARVERTICES= *variable*|*(variables)*	superimposes star at each point on median chart
WSTARCIRCLES=*n*	specifies width of STARCIRCLES= circles
WSTARS=*n*	specifies width of STARVERTICES= stars

Table 42.21. Options for Interactive Control Charts

HTML=(*variable*)	specifies a variable whose values are URLs to be associated with subgroups
HTML2=(*variable*)	specifies variable whose values are URLs to be associated with subgroups on secondary chart
HTML_LEGEND= (*variable*)	specifies a variable whose values are URLs to be associated with symbols in the symbol legend
WEBOUT=*SAS-data-set*	creates an OUTTABLE= data set with additional graphics co-ordinate data

Table 42.22. Overlay Options

CCOVERLAY=*color-list*	specifies colors for primary chart overlay line segments
CCOVERLAY2=*color-list*	specifies colors for secondary chart overlay line segments
COVERLAY=*color-list*	specifies colors for primary chart overlay plots
COVERLAY2=*color-list*	specifies colors for secondary chart overlay plots
COVERLAYCLIP=*color*	specifies color for clipped points on overlays
LOVERLAY=*linetypes*	specifies line types for primary chart overlay line segments
LOVERLAY2=*linetypes*	specifies line types for secondary chart overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=*variable-list*	specifies variables to overlay on primary chart
OVERLAY2=*variable-list*	specifies variables to overlay on secondary chart
OVERLAY2HTML=*variable-list*	specifies URLS to associate with secondary chart overlay points
OVERLAY2ID=*variable-list*	specifies labels for secondary chart overlay points
OVERLAY2SYM=*symbol-list*	specifies symbols for secondary chart overlays
OVERLAY2SYMHT=*value-list*	specifies symbol heights for secondary chart overlays
OVERLAYCLIPSYM=*symbol*	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=*value*	specifies symbol height for clipped points on overlays
OVERLAYHTML=*variable-list*	specifies URLS to associate with primary chart overlay points
OVERLAYID=*variable-list*	specifies labels for primary chart overlay points
OVERLAYLEGLAB='*label*'	specifies label for overlay legend
OVERLAYSYM=*symbol-list*	specifies symbols for primary chart overlays
OVERLAYSYMHT=*value-list*	specifies symbol heights for primary chart overlays
WOVERLAY=*value-list*	specifies widths of primary chart overlay line segments
WOVERLAY2=*value-list*	specifies widths of secondary chart overlay line segments

Details

Constructing Median Charts

The following notation is used in this section:

μ	process mean (expected value of the population of measurements)
σ	process standard deviation (standard deviation of the population of measurements)
$\bar{X}_i$	mean of measurements in i^{th} subgroup
n_i	sample size of i^{th} subgroup
N	the number of subgroups
x_{ij}	j^{th} measurement in the i^{th} subgroup, $j = 1, 2, 3, \ldots, n_i$
$x_{i(j)}$	j^{th} largest measurement in the i^{th} subgroup. Then $$x_{i(1)} \leq x_{i(2)} \leq \ldots \leq x_{i(n_i)}$$
$\overline{\overline{X}}$	weighted average of subgroup means
M_i	median of the measurements in the i^{th} subgroup: $$M_i = \begin{cases} x_{i((n_i+1)/2)} & \text{if } n_i \text{ is odd} \\ (x_{i(n_i/2)} + x_{i((n_i/2)+1)})/2 & \text{if } n_i \text{ is even} \end{cases}$$
$\bar{M}$	average of the subgroup medians: $$\bar{M} = (n_1 M_1 + \ldots + n_N M_N)/(n_1 + \ldots + n_N)$$
$\tilde{M}$	median of the subgroup medians. Denote the j^{th} largest median by $M_{(j)}$ so that $M_{(1)} \leq M_{(2)} \leq \ldots \leq M_{(N)}$. Then $$\tilde{M} = \begin{cases} M_{((N+1)/2)} & \text{if } N \text{ is odd} \\ (M_{(N/2)} + M_{(N/2)+1})/2 & \text{if } N \text{ is even} \end{cases}$$
$e_M(n)$	standard error of the median of n independent, normally distributed variables with unit standard deviation (the value of $e_M(n)$ can be calculated with the STDMED function in a DATA step)
$Q_p(n)$	$100p^{\text{th}}$ percentile $(0 < p < 1)$ of the distribution of the median of n independent observations from a normal population with unit standard deviation
z_p	$100p^{\text{th}}$ percentile of the standard normal distribution
$D_p(n)$	$100p^{\text{th}}$ percentile of the distribution of the range of n independent observations from a normal population with unit standard deviation

Plotted Points

Each point on a median chart indicates the value of a subgroup median (M_i). For example, if the tenth subgroup contains the values 12, 15, 19, 16, and 14, the value plotted for this subgroup is $M_{10} = 15$.

Central Line

The value of the central line indicates an estimate for μ, which is computed as

- $\bar{M}$ by default
- $\overline{\overline{X}}$ when you specify MEDCENTRAL=AVGMEAN
- $\tilde{M}$ when you specify MEDCENTRAL=MEDMED
- μ_0 when you specify μ_0 with the MU0= option

Control Limits

You can compute the limits

- as a specified multiple (k) of the standard error of M_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that M_i exceeds the limits

The following table provides the formulas for the limits:

Table 42.23. Limits for Median Charts

Control Limits
LCLM = lower limit = $\bar{M} - k\hat{\sigma}e_M(n_i)$ UCLM = upper limit = $\bar{M} + k\hat{\sigma}e_M(n_i)$

Probability Limits
LCLM = lower limit = $\bar{M} - Q_{\alpha/2}(n_i)\hat{\sigma}$ UCLM = upper limit = $\bar{M} + Q_{1-\alpha/2}(n_i)\hat{\sigma}$

Note that the limits vary with n_i. In Table 42.23, replace $\bar{M}$ with $\overline{\overline{X}}$ if you specify MEDCENTRAL=AVGMEAN, and replace $\bar{M}$ with $\tilde{M}$ if you specify MEDCENTRAL=MEDMED. Replace $\bar{M}$ with μ_0 if you specify μ_0 with the MU0= option, and replace $\hat{\sigma}$ with σ_0 if you specify σ_0 with the SIGMA0= option. The formulas assume that the data are normally distributed.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.

- Specify μ_0 with the MU0= option or with the variable _MEAN_ in the LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in the LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables can be saved:

Table 42.24. OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
_CP_	capability index C_p
CPK	capability index C_{pk}
CPL	capability index CPL
CPM	capability index C_{pm}
CPU	capability index CPU
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLM	lower control limit for subgroup median
LCLR	lower control limit for subgroup range
LCLS	lower control limit for subgroup standard deviation
LIMITN	sample size associated with the control limits
LSL	lower specification limit
MEAN	value of central line on median chart ($\bar{M}$, $\tilde{M}$, $\overline{\overline{X}}$, or μ_0)
_R_	value of central line on R chart
_S_	value of central line on s chart
SIGMAS	multiple (k) of standard error of M_i
STDDEV	process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*subgroup-variable* specified in the MCHART statement
TARGET	target value
TYPE	type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLM	upper control limit for subgroup median
UCLR	upper control limit for subgroup range
UCLS	upper control limit for subgroup standard deviation
USL	upper specification limit
VAR	*process* specified in the MCHART statement

Notes:

1. The variables _LCLS_, _S_, and _UCLS_ are included if you specify the STDDEVIATIONS option; otherwise, the variables _LCLR_, _R_, and _UCLR_ are included. These variables are not used to create median charts, but they allow the OUTLIMITS= data set to be used as a LIMITS= data set with the BOXCHART, XRCHART, XSCHART, and MRCHART statements.

2. If the control limits vary with subgroup sample size, the special missing value *V* is assigned to the variables _LIMITN_, _LCLM_, _UCLM_, _LCLR_, _R_, _UCLR_, _LCLS_, _S_, and _UCLS_.
3. If the limits are defined in terms of a multiple k of the standard error of M_i, the value of _ALPHA_ is computed as $\alpha = 2(1 - F_{med}(k, n))$, where $F_{med}(\cdot, n)$ is the cumulative distribution function of the median of a random sample of n standard normally distributed observations, and n is the value of _LIMITN_. If _LIMITN_ has the special missing value *V*, this value is assigned to _ALPHA_.
4. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = F^{-1}_{med}(1 - \alpha/2, n)$, where $F^{-1}_{med}(\cdot, n)$ is the inverse distribution function of the median of a random sample of n standard normally distributed observations, and n is the value of _LIMITN_. If _LIMITN_ has the special missing value *V*, this value is assigned to _SIGMAS_.
5. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you provide a target value with the TARGET= option. See "Capability Indices" on page 1774 for computational details.
6. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the MCHART statement. For an example, see "Saving Control Limits" on page 1400.

OUTHISTORY= Data Set

The OUTHISTORY= option saves subgroup summary statistics. The following variables can be saved:

- the *subgroup-variable*
- a subgroup median variable named by *process* suffixed with *M*
- a subgroup range variable named by *process* suffixed with *R*
- a subgroup standard deviation variable named by *process* suffixed with *S*
- a subgroup sample size variable named by *process* suffixed with *N*

A subgroup standard deviation variable is included if you specify the STDDEVIATIONS option; otherwise, a subgroup range variable is included.

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Variables containing subgroup summary statistics are created for each *process* specified in the MCHART statement. For example, consider the following statements:

```
proc shewhart data=steel;
   mchart (width diameter)*lot / outhistory=summary;
run;
```

The data set SUMMARY contains variables named LOT, WIDTHM, WIDTHR, WIDTHN, DIAMTERM, DIAMTERR, and DIAMTERN. The variables WIDTHR and DIAMTERR are included, since the STDDEVIATIONS option is not specified. If you specified the STDDEVIATIONS option, the data set SUMMARY would contain WIDTHS and DIAMTERS rather than WIDTHR and DIAMTERR.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 1398.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables are saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on median chart
LCLM	lower control limit for median
LIMITN	nominal sample size associated with the control limits
MEAN	estimate of process mean ($\bar{M}$, $\tilde{M}$, $\overline{\overline{X}}$, or μ_0)
SIGMAS	multiple (k) of the standard error associated with control limits
subgroup	values of the subgroup variable
SUBMED	subgroup median
SUBN	subgroup sample size
TESTS	tests for special causes signaled on median chart
UCLM	upper control limit for median
VAR	*process* specified in the MCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)
- _TREND_ (if the TRENDVAR= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The k^{th} character of a value of _TESTS_ is k if Test k is positive at that subgroup. For example, if you request all eight tests and Tests 2 and 8 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, an 8 for the eighth character, and blanks for the other six characters.
3. The variables _EXLIM_ and _TESTS_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1400.

ODS Tables

The following table summarizes the ODS tables that you can request with the MCHART statement.

Table 42.25. ODS Tables Produced with the MCHART Statement

Table Name	Description	Options
MCHART	median chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the MCHART statement must be a SAS variable in the DATA= data set. This variable provides measurements that must be grouped into subgroup samples indexed by the values of the *subgroup-variable*. The *subgroup-variable*, which is specified in the MCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. If the i^{th} subgroup contains n_i items, there should be n_i consecutive observations for which the value of the *subgroup-variable* is the index of the i^{th} subgroup. For example, if each subgroup contains five items and there are 30 subgroup samples, the DATA= data set should contain 150 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the DATA= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 1936).

For an example of a DATA= data set, see "Creating Charts for Medians from Raw Data" on page 1392.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set CONLIMS:*

```
proc shewhart data=info limits=conlims;
   mchart weight*batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLM_, _MEAN_, and _UCLM_, which specify the control limits directly
- the variables _MEAN_ and _STDDEV_, which are used to calculate the control limits according to the equations in Table 42.23 on page 1418

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are `ESTIMATE`, `STANDARD`, `STDMU`, and `STDSIGMA`.

*In Release 6.09 and in earlier releases, it is necessary to specify the READLIMITS option.

- BY variables are required if specified with a BY statement.

For an example, see the "Reading Preestablished Control Limits" section on page 1403.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This allows you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART procedure or to read output data sets created with SAS summarization procedures, such as PROC UNIVARIATE.

A HISTORY= data set used with the MCHART statement must contain the following:

- the *subgroup-variable*
- a subgroup mean variable for each *process*
- a subgroup median variable for each *process*
- a subgroup sample size variable for each *process*
- either a subgroup range variable or a subgroup standard deviation variable for each *process*

The names of the subgroup summary statistics variables must be the *process* name concatenated with the following special suffix characters:

Subgroup Summary Statistic	Suffix Character
subgroup median	M
subgroup mean	X
subgroup sample size	N
subgroup range	R
subgroup standard deviation	S

You must provide the subgroup mean variable only if you specify the MEDCENTRAL=AVGMEAN option. If you specify the STDDEVIATIONS option, the subgroup standard deviation variable must be included; otherwise, the subgroup range variable must be included.

For example, consider the following statements:

```
proc shewhart history=summary;
   mchart (weight yldstren)*batch / medcentral=avgmean;
run;
```

The data set SUMMARY must include the variables BATCH, WEIGHTX, WEIGHTM, WEIGHTR, WEIGHTN, YLDSRENX, YLDSRENM, YLDSRENR, and YLDSRENN. If the STDDEVIATIONS option were specified in the preceding MCHART statement, it would be necessary for SUMMARY to include the variables BATCH, WEIGHTX, WEIGHTM, WEIGHTS, WEIGHTN, YLDSRENX, YLDSRENM, YLDSRENS, and YLDSRENN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 1936 for an example).

For an example of a HISTORY= data set, see "Creating Charts for Medians from Subgroup Summary Data" on page 1394.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in Chapter 56, "Specialized Control Charts."

The following table lists the variables required in a TABLE= data set used with the MCHART statement:

Table 42.26. Variables Required in a TABLE= Data Set

Variable	Description
LCLM	lower control limit for median
LIMITN	nominal sample size associated with the control limits
MEAN	process mean
subgroup-variable	values of the *subgroup-variable*
SUBMED	subgroup median
SUBN	subgroup sample size
UCLM	upper control limit for median

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1400.

Methods for Estimating the Standard Deviation

When control limits are determined from the input data, three methods (referred to as default, MVLUE, and RMSDF) are available with the MCHART statement for estimating the process standard deviation σ. The method used to calculate σ depends on whether you specify the STDDEVIATIONS option in the MCHART statement. If this option is specified, σ is estimated using subgroup standard deviations; otherwise, σ is estimated using subgroup ranges. For further details and formulas, see "Methods for Estimating the Standard Deviation" on page 1723.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical	DATA=	*process*
Vertical	HISTORY=	subgroup median variable
Vertical	TABLE=	_SUBMED_

If you specify the TRENDVAR= option, you can provide distinct labels for the vertical axes of the median and trend charts by breaking the vertical axis into two parts with a split character. Specify the split character with the SPLIT= option. The first part labels the vertical axis of the median chart, and the second part labels the vertical axis of the trend chart.

For an example, see "Labeling Axes" on page 1966.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples

This section provides more advanced examples of the MCHART statement.

Example 42.1. Controlling Value of Central Line

You can specify options in the MCHART statement to request one of the following values for the central line on median charts:

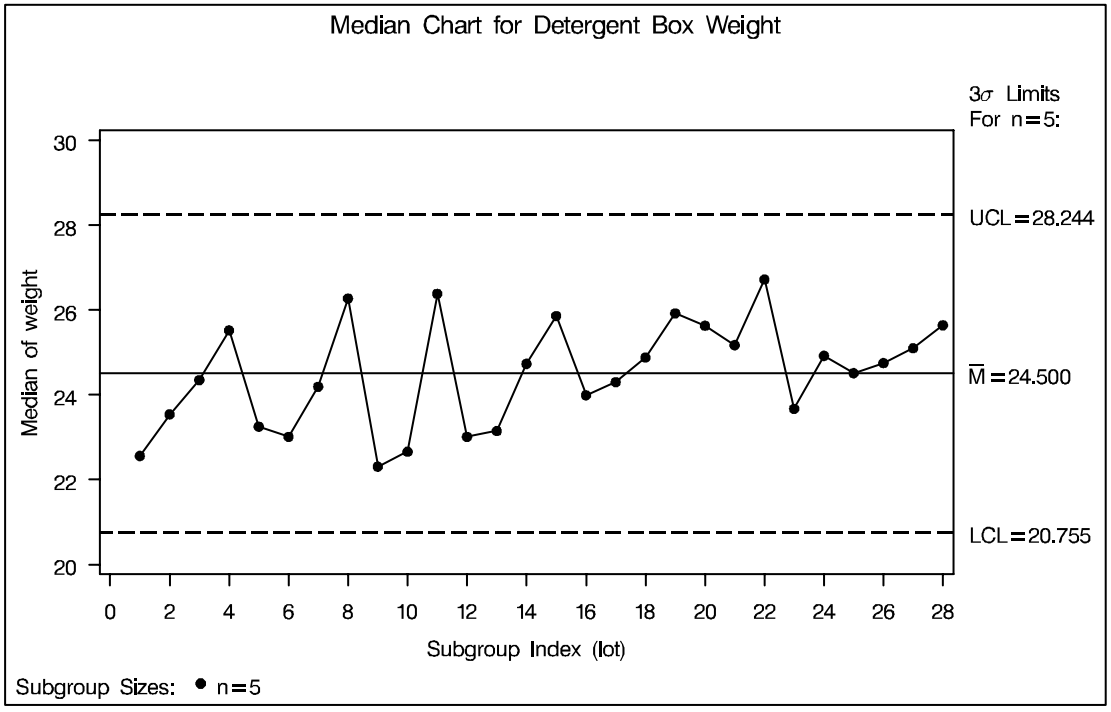
- the average of the subgroup medians
- the average of the subgroup means
- the median of the subgroup medians
- a standard value of the process mean

By default, the value of the central line is the average of the subgroup medians. The following statements create a median chart for the detergent box weights stored in the data set DETERGENT (see "Creating Charts for Medians from Raw Data" on page 1392) with the average of the subgroup medians as the central line. The resulting chart is shown in Output 42.1.1.

```
title 'Median Chart for Detergent Box Weight';
proc shewhart data=detergent;
   mchart weight*lot / ndecimal = 3;
run;
```

The NDECIMAL= option specifies the number of decimal digits in the default labels for the control limits and central line.

Output 42.1.1. Central Line is Average of Subgroup Medians

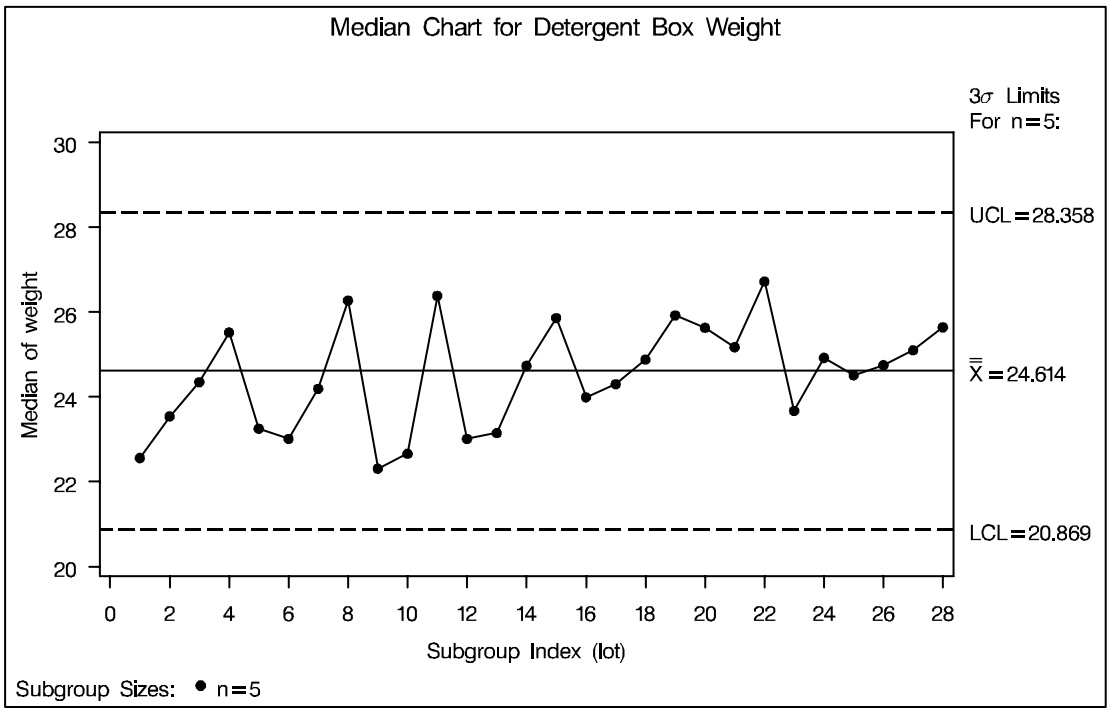


You can also request that the central line indicate the average of the subgroup means. The following statements create a median chart with this value for the central line:

```
title 'Median Chart for Detergent Box Weight';
proc shewhart data=detergent;
   mchart weight*lot / ndecimal    = 3
                       medcentral = avgmean;
run;
```

The MEDCENTRAL= option specifies the value used for the central line. In this case, MEDCENTRAL=AVGMEAN is specified to request a central line indicating the average of the subgroup means. The resulting chart is shown in Output 42.1.2.

Output 42.1.2. Central Line is Average of Subgroup Means

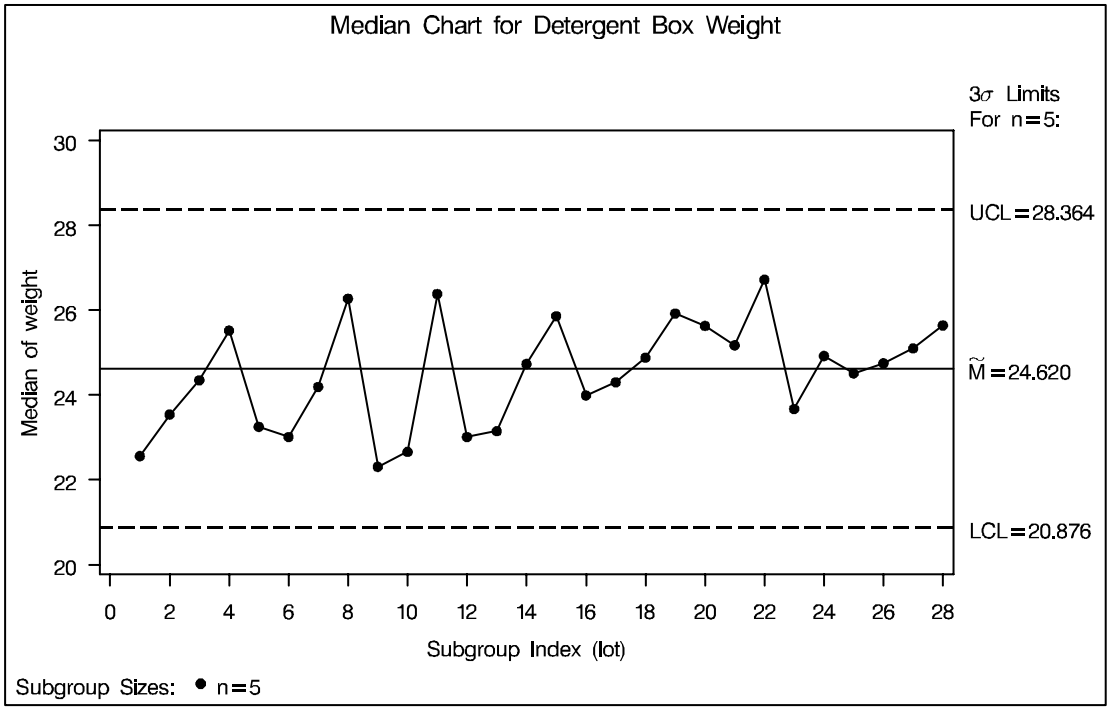


If you specify MEDCENTRAL=MEDMED, the median of the subgroup medians is used for the central line, as demonstrated by the following statements:

```
title 'Median Chart for Detergent Box Weight';
proc shewhart data=detergent;
   mchart weight*lot / ndecimal    = 3
                       medcentral = medmed;
run;
```

The resulting chart is shown in Output 42.1.3.

Output 42.1.3. Central Line is Median of Subgroup Medians

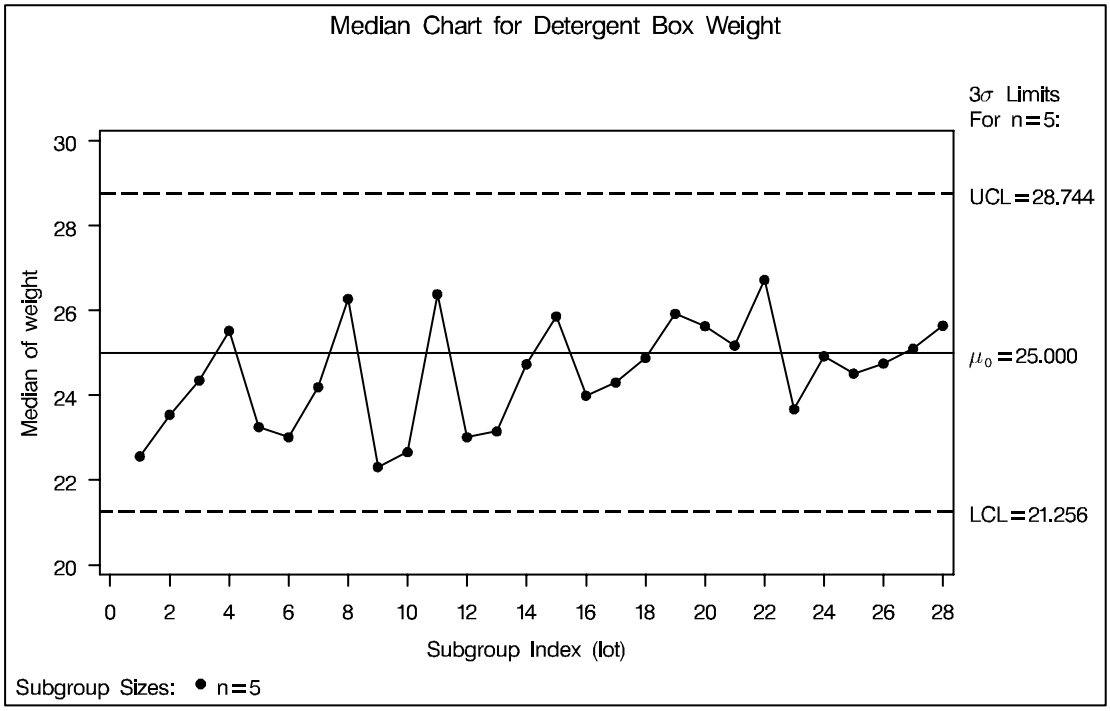


In some situations a standard value for the process mean (μ_0) is available. For instance, extensive startup testing provides an estimate of the process mean. If specified, this value is used for the central line. The following statements create a median chart for the detergent box weights with $\mu_0 = 25$:

```
title 'Median Chart for Detergent Box Weight';
proc shewhart data=detergent;
   mchart weight*lot / ndecimal = 3
                       mu0      = 25
                       xsymbol  = mu0;
run;
```

The MU0= option specifies the standard value for the process mean, and the XSYMBOL= option specifies the label for the central line. In this case, XSYMBOL=MU0 is specified to indicate that the central line represents a standard value. The resulting chart is shown in Output 42.1.4.

Output 42.1.4. Median Chart for Detergent Box Weight Data



Note that you can also provide μ_0 with the _MEAN_ variable in a LIMITS= data set. For example, the following DATA step creates a data set (DLIMS) which contains the same standard value specified in the preceding MCHART statement:

```
data dlims;
   _var_    = "WEIGHT  ";
   _subgrp_ = "LOT     ";
   _mean_   = 25;
run;
```

The _VAR_ and _SUBGRP_ variables are required if this data set is to be read as a LIMITS= data set in the PROC SHEWHART statement. These values must match the names of the *process* and *subgroup-variable* specified in the MCHART statement. The following statements specify the data set DLIMS as a LIMITS= data set and create a median chart (not shown here) identical to the one in Output 42.1.4:

```
title 'Median Chart for Detergent Box Weight';
symbol v=dot;
proc shewhart data=detergent limits=dlims;
   mchart weight*lot / xsymbol =mu0
                       ndecimal=3;
run;
```

For more information, see "Constructing Median Charts" on page 1417.

Example 42.2. Estimating the Process Standard Deviation

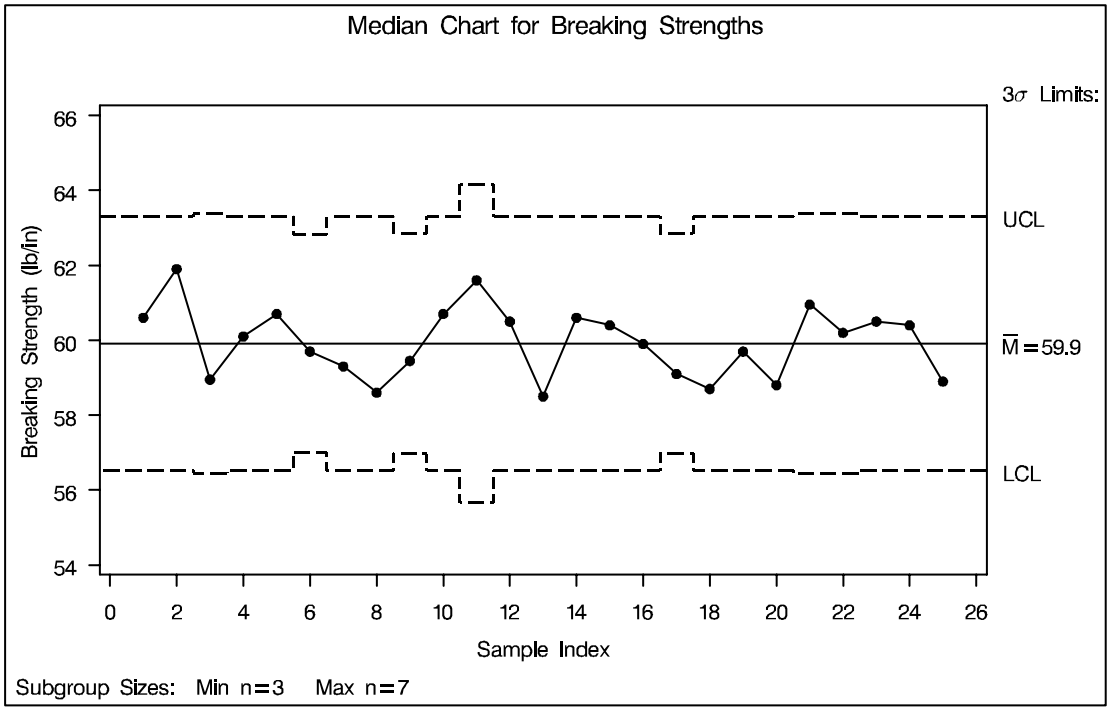
The following data set (WIRE) contains breaking strength measurements recorded in pounds per inch for 25 samples from a metal wire manufacturing process. The subgroup sample sizes vary between 3 and 7.

```
data wire;
   input sample size @;
   do i=1 to size;
      input brstr @@;
      output;
      end;
   drop i size;
   label brstr  ='Breaking Strength (lb/in)'
         sample ='Sample Index';
   datalines;
 1  5 60.6 62.3 62.0 60.4 59.9
 2  5 61.9 62.1 60.6 58.9 65.3
 3  4 57.8 60.5 60.1 57.7
 4  5 56.8 62.5 60.1 62.9 58.9
 5  5 63.0 60.7 57.2 61.0 53.5
 6  7 58.7 60.1 59.7 60.1 59.1 57.3 60.9
 7  5 59.3 61.7 59.1 58.1 60.3
 8  5 61.3 58.5 57.8 61.0 58.6
 9  6 59.5 58.3 57.5 59.4 61.5 59.6
10  5 61.7 60.7 57.2 56.5 61.5
11  3 63.9 61.6 60.9
12  5 58.7 61.4 62.4 57.3 60.5
13  5 56.8 58.5 55.7 63.0 62.7
14  5 62.1 60.6 62.1 58.7 58.3
15  5 59.1 60.4 60.4 59.0 64.1
16  5 59.9 58.8 59.2 63.0 64.9
17  6 58.8 62.4 59.4 57.1 61.2 58.6
18  5 60.3 58.7 60.5 58.6 56.2
19  5 59.2 59.8 59.7 59.3 60.0
20  5 62.3 56.0 57.0 61.8 58.8
21  4 60.5 62.0 61.4 57.7
22  4 59.3 62.4 60.4 60.0
23  5 62.4 61.3 60.5 57.7 60.2
24  5 61.2 55.5 60.2 60.4 62.4
25  5 59.0 66.1 57.7 58.5 58.9
;
```

The following statements request a median chart, shown in Output 42.2.1, for the wire breaking strength measurements:

```
title 'Median Chart for Breaking Strengths';
proc shewhart data=wire;
   mchart brstr*sample;
run;
```

Output 42.2.1. Median Chart with Varying Sample Sizes



Note that the control limits vary with the subgroup sample size. The sample size legend in the lower left corner displays the minimum and maximum subgroup sample sizes.

By default, the control limits shown in Output 42.2.1 are 3σ limits estimated from the data. You can use the STDDEVIATIONS option and the SMETHOD= option in the MCHART statement to control how the estimate of the process standard deviation σ is calculated. The STDDEVIATIONS option specifies that the estimate of σ is to be calculated from subgroup standard deviations rather than subgroup ranges, the default. The SMETHOD= option specifies the method for estimating σ. You can specify the following methods:

- NOWEIGHT
- MVLUE
- RMSDF

The NOWEIGHT method, which is the default, requests an unweighted average of subgroup estimates, the MVLUE method requests a minimum variance linear unbiased estimate, and the RMSDF method requests a weighted root-mean-square estimate. Note that the RMSDF method is only available if, in addition, you specify the STDDEVIATIONS option. For details, see "Methods for Estimating the Standard Deviation" on page 1426.

The following statements contain five MCHART statements, which calculate five different estimates for σ by specifying different combinations of options:

```
title 'Estimates of the Process Standard Deviation';
proc shewhart data=wire;
   mchart brstr*sample / outlimits=wirelim1
                         outindex ='NOWEIGHT-Ranges';
   mchart brstr*sample / outlimits=wirelim2
                         stddeviations
                         outindex ='NOWEIGHT-Stds';
   mchart brstr*sample / outlimits=wirelim3
                         smethod  =mvlue
                         outindex ='MVLUE   -Ranges';
   mchart brstr*sample / outlimits=wirelim4
                         stddeviations
                         smethod  =mvlue
                         outindex ='MVLUE   -Stds';
   mchart brstr*sample / outlimits=wirelim5
                         stddeviations
                         smethod  =rmsdf
                         outindex ='RMSDF   -Stds';
run;
```

The OUTLIMITS= option names the data set containing the control limit information. The _STDDEV_ variable in the OUTLIMITS= data set contains the estimate of the process standard deviation. The OUTINDEX= option specifies the value of the _INDEX_ variable in the OUTLIMITS= data set and is used in this example to identify the estimation method. The following statements create a data set named WLIMITS, which contains the five different estimates. This data set is listed in Output 42.2.2.

```
data wlimits;
   set wirelim1 wirelim2 wirelim3 wirelim4 wirelim5;
   keep _index_ _stddev_;
run;
```

Output 42.2.2. The Data Set WLIMITS

The WLIMITS Data Set

INDEX	_STDDEV_
NOWEIGHT-Ranges	2.11146
NOWEIGHT-Stds	2.15453
MVLUE -Ranges	2.11240
MVLUE -Stds	2.14790
RMSDF -Stds	2.17479

The median chart shown in Output 42.1.1 uses the estimate listed first in Output 42.2.2 ($\sigma = 2.11146$), since the MCHART statement used to create this chart omitted the STDDEVIATIONS option and the SMETHOD= option.

Chapter 43
MRCHART Statement

Chapter Contents

OVERVIEW 1437

GETTING STARTED 1438
- Creating Charts for Medians and Ranges from Raw Data 1438
- Creating Charts for Medians and Ranges from Summary Data 1440
- Saving Summary Statistics 1444
- Saving Control Limits 1445
- Reading Preestablished Control Limits 1447

SYNTAX 1449
- Summary of Options 1450

DETAILS 1463
- Constructing Charts for Medians and Ranges 1463
- Output Data Sets 1465
- ODS Tables 1468
- Input Data Sets 1469
- Methods for Estimating the Standard Deviation 1472
- Axis Labels 1473
- Missing Values 1473

EXAMPLES 1474
- Example 43.1. Working with Unequal Subgroup Sample Sizes 1474
- Example 43.2. Specifying Axis Labels 1478

Chapter 43
MRCHART Statement

Overview

The MRCHART statement creates charts for subgroup medians and ranges, which are used to analyze the central tendency and variability of a process.

You can use options in the MRCHART statement to

- compute control limits from the data based on a multiple of the standard error of the plotted medians and ranges or as probability limits
- tabulate subgroup sample sizes, subgroup medians, subgroup ranges, control limits, and other information
- save control limits in an output data set
- save subgroup sample sizes, subgroup medians, and subgroup ranges in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify the method for estimating the process standard deviation
- specify a known (standard) process mean and standard deviation for computing control limits
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the charts more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

Getting Started

This section introduces the MRCHART statement with simple examples that illustrate commonly used options. Complete syntax for the MRCHART statement is presented in the "Syntax" section on page 1449, and advanced examples are given in the "Examples" section on page 1474.

Creating Charts for Medians and Ranges from Raw Data

See SHWMR1 in the SAS/QC Sample Library

A consumer products company weighs detergent boxes (in pounds) to determine whether the fill process is in control. The following statements create a SAS data set named DETERGENT, which contains the weights for five boxes in each of 28 lots. A lot is considered a rational subgroup.

```
data detergent;
   input lot @;
   do i=1 to 5;
      input weight @;
      output;
   end;
drop i;
datalines;
 1 17.39 26.93 19.34 22.56 24.49
 2 23.63 23.57 23.54 20.56 22.17
 3 24.35 24.58 23.79 26.20 21.55
 4 25.52 28.02 28.44 25.07 23.39
 5 23.25 21.76 29.80 23.09 23.70
 6 23.01 22.67 24.70 20.02 26.35
 7 23.86 24.19 24.61 26.05 24.18
 8 26.00 26.82 28.03 26.27 25.85
 9 21.58 22.31 25.03 20.86 26.94
10 22.64 21.05 22.66 29.26 25.02
11 26.38 27.50 23.91 26.80 22.53
12 23.01 23.71 25.26 20.21 22.38
13 23.15 23.53 22.98 21.62 26.99
14 26.83 23.14 24.73 24.57 28.09
15 26.15 26.13 20.57 25.86 24.70
16 25.81 23.22 23.99 23.91 27.57
17 25.53 22.87 25.22 24.30 20.29
18 24.88 24.15 25.29 29.02 24.46
19 22.32 25.96 29.54 25.92 23.44
20 25.63 26.83 20.95 24.80 27.25
21 21.68 21.11 26.07 25.17 27.63
22 26.72 27.05 24.90 30.08 25.22
23 31.58 22.41 23.67 23.47 24.90
24 28.06 23.44 24.92 24.64 27.42
25 21.10 22.34 24.96 26.50 24.51
26 23.80 24.03 24.75 24.82 27.21
27 25.10 26.09 27.21 24.28 22.45
28 25.53 22.79 26.26 25.85 25.64
;
run;
```

A partial listing of DETERGENT is shown in Figure 43.1.

The Data Set DETERGENT

lot	weight
1	17.39
1	26.93
1	19.34
1	22.56
1	24.49
2	23.63
2	23.57
2	23.54
2	20.56
2	22.17
3	24.35
3	24.58
3	23.79
3	26.20
3	21.55
4	25.52
.	.
.	.
.	.

Figure 43.1. Partial Listing of the Data Set DETERGENT

The data set DETERGENT is said to be in "strung-out" form, since each observation contains the lot number and weight of a single box. The first five observations contain the weights for the first lot, the second five observations contain the weights for the second lot, and so on. Because the variable LOT classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable WEIGHT contains the weights and is referred to as the *process variable* (or *process* for short).

You can use median and range charts to determine whether the fill process is in control. The following statements create the charts shown in Figure 43.2:

```
title 'The Data Set DETERGENT';
proc print data=detergent noobs;
run;
```

This example illustrates the basic form of the MRCHART statement. After the keyword MRCHART, you specify the *process* to analyze (in this case, WEIGHT) followed by an asterisk and the *subgroup-variable* (LOT).

The input data set is specified with the DATA= option in the PROC SHEWHART statement.

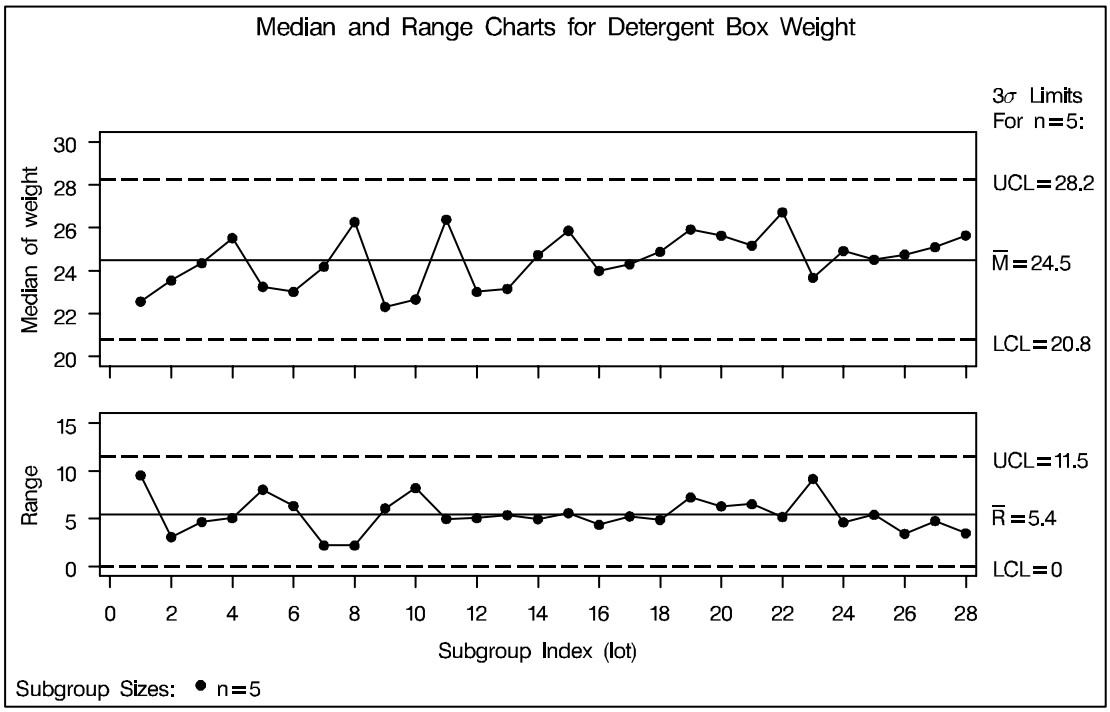


Figure 43.2. Median and Range Charts

Each point on the median chart represents the median of the measurements for a particular lot. For instance, the weights for the first lot are 17.39, 19.34, 22.56, 24.49, and 26.93, and consequently, the median plotted for this lot is 22.56. Each point on the range chart represents the range of the measurements for a particular batch. For instance, the range plotted for the first lot is $26.93 - 17.39 = 9.54$. Since all of the points lie within the control limits, you can conclude that the process is in statistical control.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 43.23 on page 1464. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1447.

For computational details, see "Constructing Charts for Medians and Ranges" on page 1463. For more details on reading raw data, see "DATA= Data Set" on page 1469.

Creating Charts for Medians and Ranges from Summary Data

See SHWMR1 in the SAS/QC Sample Library

The previous example illustrates how you can create median and range charts using raw data (process measurements). However, in many applications, the data are provided as subgroup summary statistics. This example illustrates how you can use the MRCHART statement with data of this type.

The following data set (DETSUM) provides the data from the preceding example in summarized form. There is exactly one observation for each subgroup (note that

the subgroups are still indexed by LOT). The variable WEIGHTM contains the subgroup medians, the variable WEIGHTR contains the subgroup ranges, and the variable WEIGHTN contains the subgroup sample sizes (these are all five).

```
data detsum;
   input lot weightm weightr;
   weightn = 5;
datalines;
 1  22.56  9.54
 2  23.54  3.07
 3  24.35  4.65
 4  25.52  5.05
 5  23.25  8.04
 6  23.01  6.33
 7  24.19  2.19
 8  26.27  2.18
 9  22.31  6.08
10  22.66  8.21
11  26.38  4.97
12  23.01  5.05
13  23.15  5.37
14  24.73  4.95
15  25.86  5.58
16  23.99  4.35
17  24.30  5.24
18  24.88  4.87
19  25.92  7.22
20  25.63  6.30
21  25.17  6.52
22  26.72  5.18
23  23.67  9.17
24  24.92  4.62
25  24.51  5.40
26  24.75  3.41
27  25.10  4.76
28  25.64  3.47
;
run;
```

A partial listing of DETSUM is shown in Figure 43.3.

Summary Data for Detergent Box Weights

lot	weightm	weightr	weightn
1	22.56	9.54	5
2	23.54	3.07	5
3	24.35	4.65	5
4	25.52	5.05	5
5	23.25	8.04	5
.	.	.	.
.	.	.	.
.	.	.	.

Figure 43.3. The Summary Data Set DETSUM

You can read this data set by specifying it as a HISTORY= data set in the PROC SHEWHART statement, as follows:

```
title 'Median and Range Charts for Weights';
proc shewhart history=detsum lineprinter;
   mrchart weight*lot='*';
run;
```

The charts are shown in Figure 43.4. Since the LINEPRINTER option is included in the PROC SHEWHART statement, line printer output is provided. The asterisk (*) specified in single quotes after the *subgroup-variable* indicates the character used to plot points. This character must follow an equal sign.

Note that WEIGHT is *not* the name of a SAS variable in the data set DETSUM but is, instead, the common prefix for the names of the three SAS variables WEIGHTM, WEIGHTR, and WEIGHTN. The suffix characters *M*, *R*, and *N* indicate *median*, *range*, and *sample size*, respectively. This naming convention enables you to specify three subgroup summary variables in the HISTORY= data set with a single name (WEIGHT), referred to as the *process*. The name LOT specified after the asterisk is the name of the *subgroup-variable*.

In general, a HISTORY= input data set used with the MRCHART statement must contain the following variables:

- subgroup variable
- subgroup median variable
- subgroup range variable
- subgroup sample size variable

Furthermore, the names of the subgroup median, range, and sample size variables must begin with the prefix *process* specified in the MRCHART statement and end with the special suffix characters *M*, *R*, and *N*, respectively. If the names do not follow this convention, you can use the RENAME option to rename the variables for the duration of the SHEWHART procedure step. Suppose that, instead of the variables WEIGHTM, WEIGHTR, and WEIGHTN, the data set DETSUM contained summary variables named MEDIANS, RANGES, and SIZES. The following statements would temporarily rename MEDIANS, RANGES, and SIZES to WEIGHTM, WEIGHTR, and WEIGHTN, respectively:

```
proc shewhart
   history=detsum (rename=(medians = weightm
                           ranges  = weightr
                           sizes   = weightn ));
   mrchart weight*lot;
run;
```

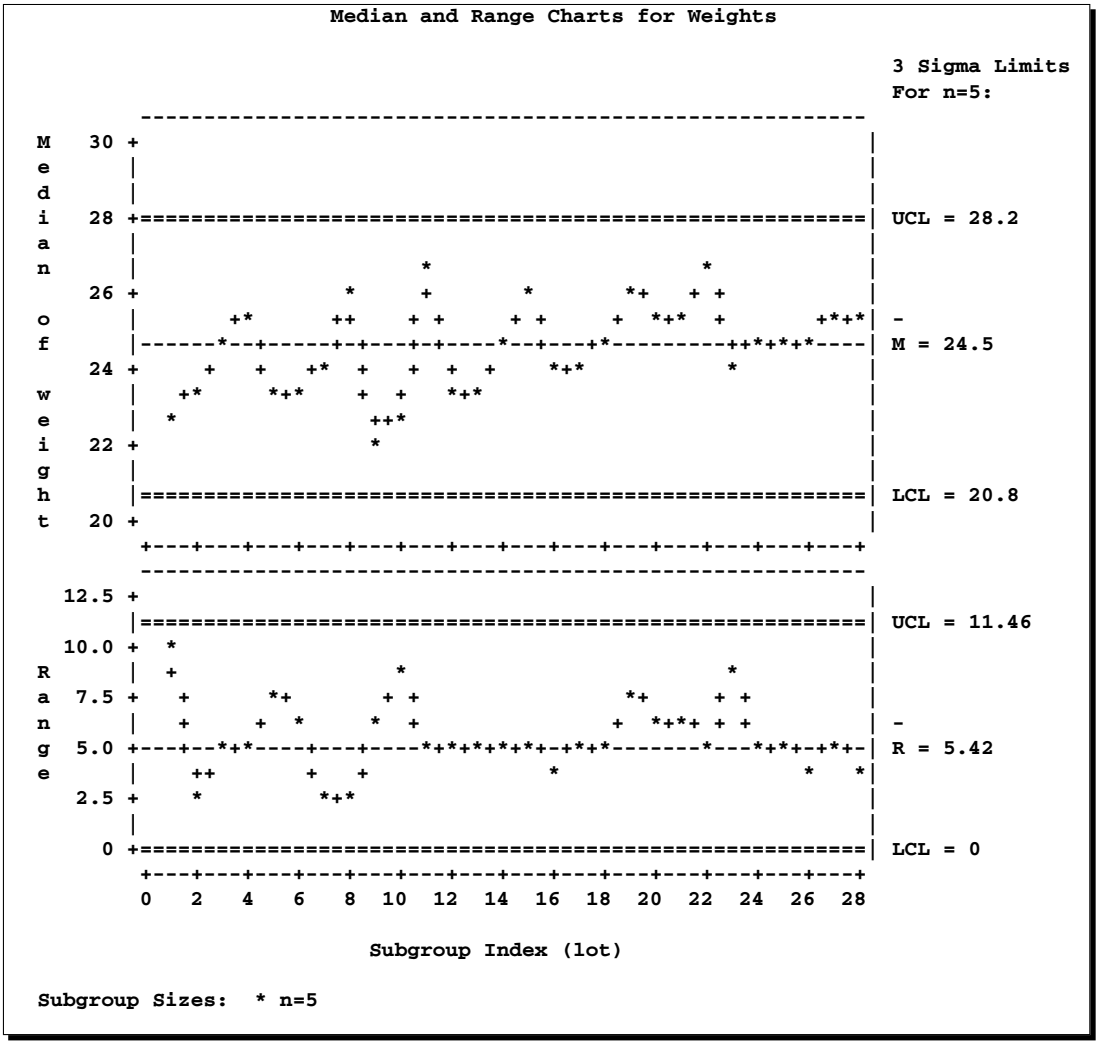


Figure 43.4. Median and Range Charts from the Summary Data Set DETSUM

In summary, the interpretation of *process* depends on the input data set:

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names containing the summary statistics.

For more information, see "HISTORY= Data Set" on page 1470.

Saving Summary Statistics

See SHWMR1 in the SAS/QC Sample Library

In this example, the MRCHART statement is used to create a summary data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read measurements from the data set DETERGENT and create a summary data set named DETHIST:

```
proc shewhart data=detergent;
   mrchart weight*lot / outhistory = dethist
                        nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the charts, which would be identical to the charts in Figure 43.2. Options such as OUTHISTORY= and NOCHART are specified after the slash (/) in the MRCHART statement. A complete list of options is presented in the "Syntax" section on page 1449.

Figure 43.5 contains a partial listing of DETHIST.

Summary Data Set DETHIST for Detergent Box Weights

lot	weight M	weight R	weight N
1	22.56	9.54	5
2	23.54	3.07	5
3	24.35	4.65	5
4	25.52	5.05	5
5	23.25	8.04	5
.	.	.	.
.	.	.	.
.	.	.	.

Figure 43.5. The Summary Data Set DETHIST

There are four variables in the data set DETHIST.

- LOT contains the subgroup index.
- WEIGHTM contains the subgroup medians.
- WEIGHTR contains the subgroup ranges.
- WEIGHTN contains the subgroup sample sizes.

Note that the summary statistic variables are named by adding the suffix characters *M*, *R*, and *N* to the *process* WEIGHT specified in the MRCHART statement. In other

words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 1466.

Saving Control Limits

See SHWMR1 in the SAS/QC Sample Library

You can save the control limits for median and range charts in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1447) or modify the limits with a DATA step program.

The following statements read measurements from the data set DETERGENT (see page 1438) and save the control limits displayed in Figure 43.2 in a data set named DETLIM:

```
proc shewhart data=detergent;
   mrchart weight*lot / outlimits=detlim
                        nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the charts. The data set DETLIM is listed in Figure 43.6.

Control Limits for Detergent Box Weights

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLM_
weight	lot	ESTIMATE	5	.002909021	3	20.7554

MEAN	_UCLM_	_LCLR_	_R_	_UCLR_	_STDDEV_
24.4996	28.2439	0	5.42036	11.4613	2.33041

Figure 43.6. The Data Set DETLIM Containing Control Limit Information

The data set DETLIM contains one observation with the limits for *process* WEIGHT. The variables _LCLM_ and _UCLM_ contain the control limits for the medians, and the variable _MEAN_ contains the central line. The variables _LCLR_ and _UCLR_ contain the control limits for the ranges, and the variable _R_ contains the central line. The values of _MEAN_ and _STDDEV_ are estimates of the process mean and process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values. For more information, see "OUTLIMITS= Data Set" on page 1465.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=detergent;
   mrchart weight*lot / outtable=dtable
                        nochart;
run;
```

This data set contains one observation for each subgroup sample. The variables _SUBMED_, _SUBR_, and _SUBN_ contain the subgroup medians, subgroup ranges, and subgroup sample sizes. The variables _LCLM_ and _UCLM_ contain the control limits for the median chart, and the variables _LCLR_ and _UCLR_ contain the control limits for the range chart. The variable _MEAN_ contains the central line for the median chart, and the variable _R_ contains the central line for the range chart. The variables _VAR_ and BATCH contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1467.

The data set DTABLE is listed in Figure 43.7.

Summary Statistics and Control Limit Information

VAR	lot	_SIGMA_	_LIMSUBN_	_LCLM_	_SUBMED_	_UCLM_	_MEAN_	_EXLCLR_	_SUBR_	_R_	_UCLR_	_EXLIMR_
weight	1	3	5	5	20.7554	22.56	24.4996	28.2439	0	9.54	5.42036	11.4613
weight	2	3	5	5	20.7554	23.54	24.4996	28.2439	0	3.07	5.42036	11.4613
weight	3	3	5	5	20.7554	24.35	24.4996	28.2439	0	4.65	5.42036	11.4613
weight	4	3	5	5	20.7554	25.52	24.4996	28.2439	0	5.05	5.42036	11.4613
weight	5	3	5	5	20.7554	23.25	24.4996	28.2439	0	8.04	5.42036	11.4613
weight	6	3	5	5	20.7554	23.01	24.4996	28.2439	0	6.33	5.42036	11.4613
weight	7	3	5	5	20.7554	24.19	24.4996	28.2439	0	2.19	5.42036	11.4613
weight	8	3	5	5	20.7554	26.27	24.4996	28.2439	0	2.18	5.42036	11.4613
weight	9	3	5	5	20.7554	22.31	24.4996	28.2439	0	6.08	5.42036	11.4613
weight	10	3	5	5	20.7554	22.66	24.4996	28.2439	0	8.21	5.42036	11.4613
weight	11	3	5	5	20.7554	26.38	24.4996	28.2439	0	4.97	5.42036	11.4613
weight	12	3	5	5	20.7554	23.01	24.4996	28.2439	0	5.05	5.42036	11.4613
weight	13	3	5	5	20.7554	23.15	24.4996	28.2439	0	5.37	5.42036	11.4613
weight	14	3	5	5	20.7554	24.73	24.4996	28.2439	0	4.95	5.42036	11.4613
weight	15	3	5	5	20.7554	25.86	24.4996	28.2439	0	5.58	5.42036	11.4613
weight	16	3	5	5	20.7554	23.99	24.4996	28.2439	0	4.35	5.42036	11.4613
weight	17	3	5	5	20.7554	24.30	24.4996	28.2439	0	5.24	5.42036	11.4613
weight	18	3	5	5	20.7554	24.88	24.4996	28.2439	0	4.87	5.42036	11.4613
weight	19	3	5	5	20.7554	25.92	24.4996	28.2439	0	7.22	5.42036	11.4613
weight	20	3	5	5	20.7554	25.63	24.4996	28.2439	0	6.30	5.42036	11.4613
weight	21	3	5	5	20.7554	25.17	24.4996	28.2439	0	6.52	5.42036	11.4613
weight	22	3	5	5	20.7554	26.72	24.4996	28.2439	0	5.18	5.42036	11.4613
weight	23	3	5	5	20.7554	23.67	24.4996	28.2439	0	9.17	5.42036	11.4613
weight	24	3	5	5	20.7554	24.92	24.4996	28.2439	0	4.62	5.42036	11.4613
weight	25	3	5	5	20.7554	24.51	24.4996	28.2439	0	5.40	5.42036	11.4613
weight	26	3	5	5	20.7554	24.75	24.4996	28.2439	0	3.41	5.42036	11.4613
weight	27	3	5	5	20.7554	25.10	24.4996	28.2439	0	4.76	5.42036	11.4613
weight	28	3	5	5	20.7554	25.64	24.4996	28.2439	0	3.47	5.42036	11.4613

Figure 43.7. The Data Set DTABLE

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read DTABLE and display charts (not shown here) identical to those in Figure 43.2:

```
title 'Median and Range Charts for Detergent Box Weight';
proc shewhart table=dtable;
   mrchart weight*lot;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see Chapter 56, "Specialized Control Charts,"). For more information, see "TABLE= Data Set" on page 1471.

Reading Preestablished Control Limits

In the previous example, the OUTLIMITS= data set DETLIM saved control limits computed from the measurements in DETERGENT. This example shows how these limits can be applied to new data provided in the following data set:

See SHWMR1 in the SAS/QC Sample Library

```
data detergt2;
   input lot @;
   do i=1 to 5;
      input weight @;
      output;
      end;
   drop i;
   datalines;
29 16.66 27.49 18.87 22.53 24.72
30 23.74 23.67 23.64 20.26 22.09
31 24.56 24.82 23.92 26.67 21.38
32 25.89 28.73 29.21 25.38 23.47
33 23.32 21.61 30.75 23.13 23.82
34 23.04 22.65 24.96 19.64 26.84
35 24.01 24.38 24.86 26.50 24.37
36 26.43 27.36 28.74 26.74 26.27
37 21.41 22.24 25.34 20.59 27.51
38 22.62 20.81 22.64 30.15 25.32
39 26.86 28.14 24.06 27.35 22.49
40 23.03 23.83 25.59 19.85 22.33
41 23.19 23.63 23.00 21.46 27.57
42 27.38 23.18 24.99 24.81 28.82
43 26.60 26.58 20.26 26.27 24.96
44 26.22 23.28 24.15 24.06 28.23
45 25.90 22.88 25.55 24.50 19.95
46 16.66 27.49 18.87 22.53 24.72
47 23.74 23.67 23.64 20.26 22.09
48 24.56 24.82 23.92 26.67 21.38
49 25.89 28.73 29.21 25.38 23.47
50 23.32 21.61 30.75 23.13 23.82
;
run;
```

The following statements create median and range charts for the data in DETERGT2 using the control limits in DETLIM:

```
symbol h = .8;
title 'Median and Range Charts for Weights';
proc shewhart data=detergt2 limits=detlim;
   mrchart weight*lot;
run;
```

The charts are shown in Figure 43.8. The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default,* this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name WEIGHT
- the value of _SUBGRP_ matches the *subgroup-variable* name LOT

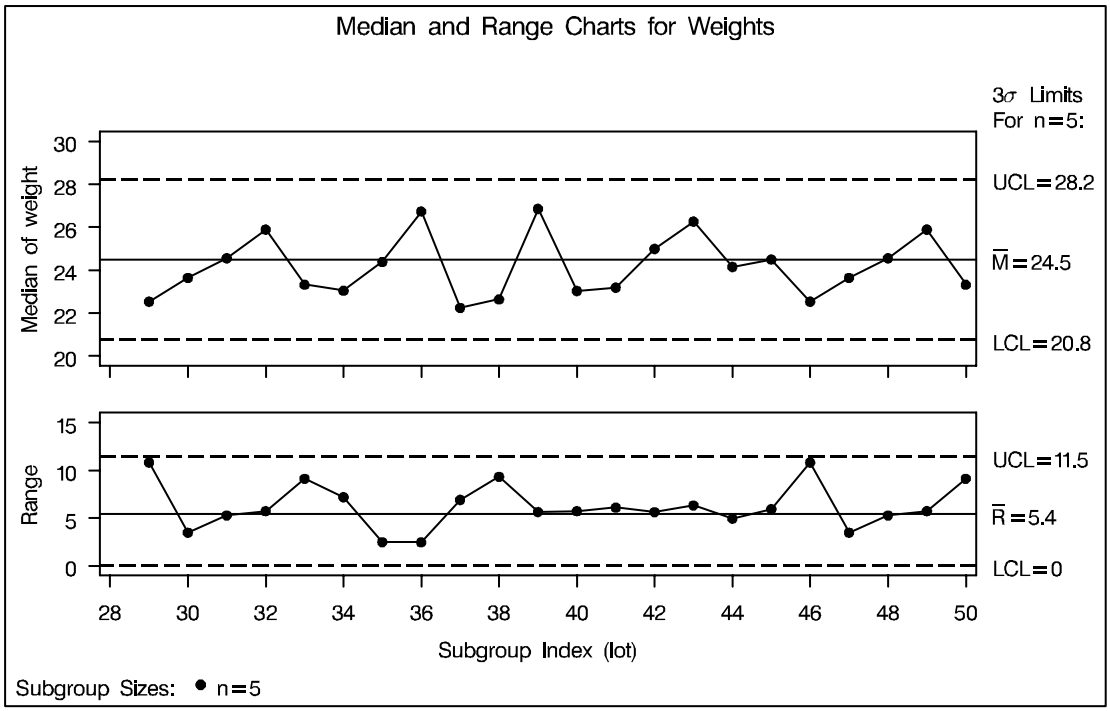


Figure 43.8. Median and Range Charts for Second Set of Detergent Box Weights

The charts indicate that the process is in control, since all the medians and ranges lie within the control limits.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1469 for details concerning the variables that you must provide.

*In Release 6.09 and in earlier releases, it is also necessary to specify the READLIMITS option to read control limits from a LIMITS= data set.

Syntax

The basic syntax for the MRCHART statement is as follows:

MRCHART *process***subgroup-variable* **;**

The general form of this syntax is as follows:

MRCHART (*processes***)****subgroup-variable* <**(***block-variables* **)** >
< =*symbol-variable* | =*'character'* > < **/** *options* >**;**

You can use any number of MRCHART statements in the SHEWHART procedure. The components of the MRCHART statement are described as follows.

process
processes
identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating Charts for Medians and Ranges from Raw Data" on page 1438.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating Charts for Medians and Ranges from Summary Data" on page 1440.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1445.

A *process* is required. If you specify more than one *process*, enclose the list in parentheses. For example, the following statements request distinct median and range charts for WEIGHT, LENGTH, and WIDTH:

```
proc shewhart data=measures;
   mrchart (weight length width)*day;
run;
```

subgroup-variable
is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding MRCHART statement, DAY is the subgroup variable. For details, see "Subgroup Variables" on page 1771.

block-variables
are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 1932 for an example.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the medians and ranges.

- If you produce a chart on a line printer, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce a chart on a graphics device, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOLn statements. See "Displaying Stratification in Levels of a Classification Variable" on page 1931 for an example.

character
specifies a plotting character for charts produced on line printers. For example, the following statements create median and range charts using an asterisk (*) to plot the points:

```
proc shewhart data=values;
   mrchart weight*day='*';
run;
```

options
enhance the appearance of the charts, request additional analyses, save results in data sets, and so on. The "Summary of Options" section, which follows, lists all options by function. Chapter 53, "Dictionary of Options," describes each option in detail.

Summary of Options

The following tables list the MRCHART statement options by function. For complete descriptions, see Chapter 53, "Dictionary of Options."

Table 43.1. Tabulation Options

TABLE	creates a basic table of subgroup medians, subgroup ranges, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUT, and TABLETESTS
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with a column indicating which tests for special causes are positive

Note that specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.

Table 43.2. Options for Specifying Tests for Special Causes

NO3SIGMACHECK	allows tests to be applied with control limits other than 3σ limits
TESTS=*value-list*| *customized-pattern-list*	specifies tests for special causes for the median chart
TESTS2=*value-list*| *customized-pattern-list*	specifies tests for special causes for the range chart
TEST2RESET=*variable*	allows tests for special causes to be reset for the range chart
TEST2RUN=*n*	specifies length of pattern for Test 2
TEST3RUN=*n*	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL='*label*'| (*variable*)|*keyword*	provides labels for points where test is positive
TESTLABEL*n*='*label*'	specifies label for n^{th} test for special causes
TESTNMETHOD= STANDARDIZE	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
TESTRESET=*variable*	allows tests for special causes to be reset for the median chart
ZONELABELS	adds labels A, B, and C to zone lines for median chart
ZONE2LABELS	adds labels A, B, and C to zone lines for range chart
ZONES	adds lines to median chart delineating zones A, B, and C
ZONES2	adds lines to range chart delineating zones A, B, and C
ZONEVALPOS=*n*	specifies position of ZONEVALUES and ZONE2VALUES labels
ZONEVALUES	labels median chart zone lines with their values
ZONE2VALUES	labels range chart zone lines with their values

Table 43.3. Graphical Options for Displaying Tests for Special Causes

CTESTLABBOX=*color*	specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=*color*| *test-color-list*	specifies color for labels used to identify points where test is positive
CTESTSYMBOL=*color*	specifies color for symbol used to plot points where test is positive
CZONES=*color*	specifies color for lines and labels delineating zones A, B, and C
LTESTS=*linetype*	specifies type of line connecting points where test is positive
LZONES=*linetype*	specifies line type for lines delineating zones A, B, and C
TESTFONT=*font*	specifies software font for labels at points where test is positive
TESTHEIGHT=*value*	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=*symbol*	specifies plot symbol for points where test is positive
TESTSYMBOLHT=*value*	specifies symbol height for points where test is positive
WTESTS=*n*	specifies width of line connecting points where test is positive

Table 43.4. Line Printer Options for Displaying Tests for Special Causes

TESTCHAR='*character*'	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
ZONECHAR='*character*'	specifies character for lines that delineate zones for tests for special causes

Table 43.5. Reference Line Options

CHREF=*color*	specifies color for lines requested by HREF= and HREF2= options
CVREF=*color*	specifies color for lines requested by VREF= and VREF2= options
HREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on median chart
HREF2=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on range chart
HREFCHAR='*character*'	specifies line character for HREF= and HREF2= lines
HREFDATA= *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on median chart
HREF2DATA= *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on range chart
HREFLABELS= *'label1'...'labeln'*	specifies labels for HREF= lines
HREF2LABELS= *'label1'...'labeln'*	specifies labels for HREF2= lines
HREFLABPOS=*n*	specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=*linetype*	specifies line type for HREF= and HREF2= lines
LVREF=*linetype*	specifies line type for VREF= and VREF2= lines
NOBYREF	specifies that reference line information in a data set is to be applied uniformly to charts created for all BY groups
VREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to vertical axis on median chart
VREF2=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to vertical axis on range chart
VREFCHAR='*character*'	specifies line character for VREF= and VREF2= lines
VREFLABELS= *'label1'...'labeln'*	specifies labels for VREF= lines
VREF2LABELS= *'label1'...'labeln'*	specifies labels for VREF2= lines
VREFLABPOS=*n*	specifies position of VREFLABELS= and VREF2LABELS= labels

Table 43.6. Clipping Options

CCLIP=*color*	specifies color for plot symbol for clipped points
CLIPCHAR='*character*'	specifies plot character for clipped points
CLIPFACTOR=*value*	determines extent to which extreme points are clipped
CLIPLEGEND='*string*'	specifies text for clipping legend
CLIPLEGPOS=*keyword*	specifies position of clipping legend
CLIPSUBCHAR='*character*'	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=*symbol*	specifies plot symbol for clipped points
CLIPSYMBOLHT=*value*	specifies symbol marker height for clipped points

Table 43.7. Axis and Axis Label Options

CAXIS=*color*	specifies color for axis lines and tick marks
CFRAME=*color*|(*color-list*)	specifies fill colors for frame for plot area
CTEXT=*color*	specifies color for tick mark values and axis labels
HAXIS=*values*|AXIS*n*	specifies major tick mark values for horizontal axis
HEIGHT=*value*	specifies height of axis label and axis legend text
HMINOR=*n*	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=*value*	specifies length of offset at both ends of horizontal axis
INTSTART=*value*	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	suppresses vertical axis truncation at zero applied by default to range chart
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=*n*	specifies thinning factor for tick mark labels on horizontal axis
SPLIT='*character*'	specifies splitting character for axis labels
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=*values*|AXIS*n*	specifies major tick mark values for vertical axis of median chart
VAXIS2=*values*|AXIS*n*	specifies major tick mark values for vertical axis of range chart
VFORMAT=*format*	specifies format for primary vertical axis tick mark labels
VFORMAT2=*format*	specifies format for secondary vertical axis tick mark labels
VMINOR=*n*	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=*value*	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis for primary chart
VZERO2	forces origin to be included in vertical axis for secondary chart
WAXIS=*n*	specifies width of axis lines

Table 43.8. Options for Plotting and Labeling Points

ALLLABEL=VALUE|(*variable*)	labels every point on median chart
ALLLABEL2=VALUE|(*variable*)	labels every point on range chart
CLABEL=*color*	specifies color for labels
CCONNECT=*color*	specifies color for line segments that connect points on chart
CFRAMELAB=*color*	specifies fill color for frame around labeled points
CNEEDLES=*color*	specifies color for needles that connect points to central line
CONNECTCHAR='*character*'	specifies character used to form line segments that connect points on chart
COUT=*color*	specifies color for portions of line segments that connect points outside control limits
COUTFILL=*color*	specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=*angle*	specifies angle at which labels are drawn
LABELFONT=*font*	specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=*value*	specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on chart
OUTLABEL=VALUE|(*variable*)	labels points outside control limits on median chart
OUTLABEL2=VALUE|(*variable*)	labels points outside control limits on range chart
SYMBOLCHARS='*characters*'	specifies characters indicating *symbol-variable*
SYMBOLLEGEND=NONE|*name*	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=*keyword*	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=*n*	specifies width of needles

Table 43.9. Options for Specifying Control Limits

ALPHA=*value*	requests probability limits for control charts
LIMITN=*n*|VARYING	specifies either nominal sample size for fixed control limits or varying control limits
NOREADLIMITS	computes control limits for each *process* from the data rather than from a LIMITS= data set (Release 6.10 and later releases)
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEXES=ALL| '*label1*'...'*labeln*'	reads multiple sets of control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (Release 6.09 and earlier releases)
SIGMAS=*k*	specifies width of control limits in terms of multiple *k* of standard error of plotted statistic

Table 43.10. Options for Displaying Control Limits

CINFILL=*color*	specifies color for area inside control limits
CLIMITS=*color*	specifies color of control limits, central line, and related labels
LCLLABEL='*label*'	specifies label for lower control limit on median chart
LCLLABEL2='*label*'	specifies label for lower control limit on range chart
LIMLABSUBCHAR='*character*'	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=*linetype*	specifies line type for control limits
NDECIMAL=*n*	specifies number of digits to right of decimal place in default labels for control limits and central line on median chart
NDECIMAL2=*n*	specifies number of digits to right of decimal place in default labels for control limits and central line on range chart
NOCTL	suppresses display of central line on median chart
NOCTL2	suppresses display of central line on range chart
NOLCL	suppresses display of lower control limit on median chart
NOLCL2	suppresses display of lower control limit on range chart
NOLIMITLABEL	suppresses labels for control limits and central lines
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOLIMIT0	suppresses display of zero lower control limit on range chart
NOUCL	suppresses display of upper control limit on median chart
NOUCL2	suppresses display of upper control limit on range chart
RSYMBOL='*string*'|*keyword*	specifies label for central line on range chart
UCLLABEL='*string*'	specifies label for upper control limit on median chart
UCLLABEL2='*string*'	specifies label for upper control limit on range chart
WLIMITS=*n*	specifies width for control limits and central line
XSYMBOL='*string*'|*keyword*	specifies label for central line on median chart

Table 43.11. Grid Options

CGRID=*color*	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=*linetype*	specifies line type for grid requested with the ENDGRID option
LGRID=*linetype*	specifies line type for grid requested with the GRID option
WGRID=*n*	specifies width of grid lines

Table 43.12. Phase Options

CPHASELEG=*color*	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE='*string*'	specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=*value*| *keyword*	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES= ALL| '*label1*'...'*labeln*'	specifies *phases* to be read from an input data set

Table 43.13. Specification Limit Options

CIINDICES (ALPHA=*value* TYPE=*keyword*)	specifies α value and type for computing capability index confidence limits
LSL=*value-list*	specifies list of lower specification limits
TARGET=*value-list*	specifies list of target values
USL=*value-list*	specifies list of upper specification limits

Table 43.14. Process Mean and Standard Deviation Options

MEDCENTRAL=*keyword*	specifies method for estimating process mean μ
MU0=*value*	specifies known value μ_0 for process mean μ
SIGMA0=*value*	specifies known value σ_0 for process standard deviation σ
SMETHOD=*keyword*	specifies method for estimating process standard deviation σ
TYPE=*keyword*	identifies whether parameters are estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set

Table 43.15. Input Data Set Options

MISSBREAK	specifies that observations with missing values are not to be processed

Table 43.16. Output Data Set Options

OUTHISTORY= *SAS-data-set*	creates output data set containing subgroup summary statistics
OUTINDEX='*string*'	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS= *SAS-data-set*	creates output data set containing control limits
OUTTABLE= *SAS-data-set*	creates output data set containing subgroup summary statistics and control limits

Table 43.17. Plot Layout Options

ALLN	plots summary statistics for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts for a process variable only when exceptions occur
INTERVAL=*keyword*	specifies natural time interval between consecutive subgroup positions when time, date, or datetime format is used
MAXPANELS=*n*	specifies maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of charts
NOCHART2	suppresses creation of range chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=*n*	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
SEPARATE	displays median and range charts on separate screens or pages
TOTPANELS=*n*	specifies number of pages or screens to be used to display chart
YPCT1=*value*	specifies length of vertical axis on median chart as a percentage of sum of lengths of vertical axes for median and range charts
ZEROSTD	displays median and range chart regardless of whether $\hat{\sigma} = 0$

Table 43.18. Graphical Enhancement Options

ANNOTATE=*SAS-data-set*	specifies annotate data set that adds features to median chart
ANNOTATE2=*SAS-data-set*	specifies annotate data set that adds features to range chart
DESCRIPTION='*string*'	specifies string that appears in the description field of the PROC GREPLAY master menu for median chart
DESCRIPTION2='*string*'	specifies string that appears in the description field of the PROC GREPLAY master menu for range chart
FONT=*font*	specifies software font for labels and legends on charts
NAME='*string*'	specifies name that appears in the name field of the PROC GREPLAY master menu for median chart
NAME2='*string*'	specifies name that appears in the name field of the PROC GREPLAY master menu for range chart
PAGENUM='*string*'	specifies the form of the label used in pagination
PAGENUMPOS= *keyword*	specifies the position of the page number requested with the PAGENUM= option

Table 43.19. Star Options

CSTARCIRCLES=*color*	specifies color for STARCIRCLES= circles
CSTARFILL=*color*|*(variable)*	specifies color for filling stars
CSTAROUT=*color*	specifies outline color for stars exceeding inner or outer circles
CSTARS=*color*| *(variable)*	specifies color for outlines of stars
LSTARCIRCLES= *linetypes*	specifies line types for STARCIRCLES= circles
LSTARS=*linetype*|*(variable)*	specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=*value*	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=*value-list*	specifies reference circles for stars
STARINRADIUS=*value*	specifies inner radius of stars
STARLABEL=*keyword*	specifies vertices to be labeled
STARLEGEND=*keyword*	specifies style of legend for star vertices
STARLEGENDLAB=*'label'*	specifies label for STARLEGEND= legend
STAROUTRADIUS=*value*	specifies outer radius of stars
STARSPEC=*value*|*SAS-data-set*	specifies method used to standardize vertex variables
STARSTART=*value*	specifies angle for first vertex
STARTYPE=*keyword*	specifies graphical style of star
STARVERTICES=*variable*|*(variables)*	superimposes star at each point on median chart
WSTARCIRCLES=*n*	specifies width of STARCIRCLES= circles
WSTARS=*n*	specifies width of STARVERTICES= stars

Table 43.20. Block Variable Legend Options

BLOCKLABELPOS=*keyword*	specifies position of label for *block-variable* legend
BLOCKLABTYPE=*value*|*keyword*	specifies text size of *block-variable* legend
BLOCKPOS=*n*	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=*color-list*	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=*variable*|*(variables)*	specifies one or more variables whose values are colors for filling background of *block-variable* legend

Table 43.21. Options for Interactive Control Charts

HTML=(*variable*)	specifies a variable whose values are URLs to be associated with subgroups
HTML2=(*variable*)	specifies variable whose values are URLs to be associated with subgroups on secondary chart
HTML_LEGEND=(*variable*)	specifies a variable whose values are URLs to be associated with symbols in the symbol legend
WEBOUT=*SAS-data-set*	creates an OUTTABLE= data set with additional graphics co-ordinate data

Table 43.22. Overlay Options

CCOVERLAY=*color-list*	specifies colors for primary chart overlay line segments
CCOVERLAY2=*color-list*	specifies colors for secondary chart overlay line segments
COVERLAY=*color-list*	specifies colors for primary chart overlay plots
COVERLAY2=*color-list*	specifies colors for secondary chart overlay plots
COVERLAYCLIP=*color*	specifies color for clipped points on overlays
LOVERLAY=*linetypes*	specifies line types for primary chart overlay line segments
LOVERLAY2=*linetypes*	specifies line types for secondary chart overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=*variable-list*	specifies variables to overlay on primary chart
OVERLAY2=*variable-list*	specifies variables to overlay on secondary chart
OVERLAY2HTML=*variable-list*	specifies URLS to associate with secondary chart overlay points
OVERLAY2ID=*variable-list*	specifies labels for secondary chart overlay points
OVERLAY2SYM=*symbol-list*	specifies symbols for secondary chart overlays
OVERLAY2SYMHT=*value-list*	specifies symbol heights for secondary chart overlays
OVERLAYCLIPSYM=*symbol*	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=*value*	specifies symbol height for clipped points on overlays
OVERLAYHTML=*variable-list*	specifies URLS to associate with primary chart overlay points
OVERLAYID=*variable-list*	specifies labels for primary chart overlay points
OVERLAYLEGLAB='*label*'	specifies label for overlay legend
OVERLAYSYM=*symbol-list*	specifies symbols for primary chart overlays
OVERLAYSYMHT=*value-list*	specifies symbol heights for primary chart overlays
WOVERLAY=*value-list*	specifies widths of primary chart overlay line segments
WOVERLAY2=*value-list*	specifies widths of secondary chart overlay line segments

Details

Constructing Charts for Medians and Ranges

The following notation is used in this section:

μ	process mean (expected value of the population of measurements)
σ	process standard deviation (standard deviation of the population of measurements)
$\bar{X}_i$	mean of measurements in i^{th} subgroup
R_i	range of measurements in i^{th} subgroup
n_i	sample size of i^{th} subgroup
N	the number of subgroups
x_{ij}	j^{th} measurement in the i^{th} subgroup, $j = 1, 2, 3, \ldots, n_i$
$x_{i(j)}$	j^{th} largest measurement in the i^{th} subgroup. Then $$x_{i(1)} \leq x_{i(2)} \leq \ldots \leq x_{i(n_i)}$$
$\overline{\overline{X}}$	weighted average of subgroup means
M_i	median of the measurements in the i^{th} subgroup: $$M_i = \begin{cases} x_{i((n_i+1)/2)} & \text{if } n_i \text{ is odd} \\ (x_{i(n_i/2)} + x_{i((n_i/2)+1)})/2 & \text{if } n_i \text{ is even} \end{cases}$$
$\bar{M}$	average of the subgroup medians: $$\bar{M} = (n_1 M_1 + \ldots + n_N M_N)/(n_1 + \ldots + n_N)$$
$\tilde{M}$	median of the subgroup medians. Denote the j^{th} largest median by $M_{(j)}$ so that $M_{(1)} \leq M_{(2)} \leq \ldots \leq M_{(N)}$. $$\tilde{M} = \begin{cases} M_{((N+1)/2)} & \text{if } N \text{ is odd} \\ (M_{(N/2)} + M_{(N/2)+1})/2 & \text{if } N \text{ is even} \end{cases}$$
$e_M(n)$	standard error of the median of n independent, normally distributed variables with unit standard deviation (the value of $e_M(n)$ can be calculated with the STDMED function in a DATA step)
$Q_p(n)$	$100p^{\text{th}}$ percentile $(0 < p < 1)$ of the distribution of the median of n independent observations from a normal population with unit standard deviation
$d_2(n)$	expected value of the range of n independent normally distributed variables with unit standard deviation
$d_3(n)$	standard error of the range of n independent observations from a normal population with unit standard deviation
z_p	$100p^{\text{th}}$ percentile of the standard normal distribution
$D_p(n)$	$100p^{\text{th}}$ percentile of the distribution of the range of n independent observations from a normal population with unit standard deviation

Plotted Points

Each point on a median chart indicates the value of a subgroup median (M_i). For example, if the tenth subgroup contains the values 12, 15, 19, 16, and 14, the value plotted for this subgroup is $M_{10} = 15$. Each point on a range chart indicates the value of a subgroup range (R_i). For example, the value plotted for the tenth subgroup is $R_{10} = 19 - 12 = 7$.

Central Lines

On a median chart, the value of the central line indicates an estimate for μ, which is computed as

- $\bar{M}$ by default
- $\overline{\overline{X}}$ when you specify MEDCENTRAL=AVGMEAN
- $\tilde{M}$ when you specify MEDCENTRAL=MEDMED
- μ_0 when you specify μ_0 with the MU0= option

On the range chart, by default, the central line for the i^{th} subgroup indicates an estimate for the expected value of R_i, which is computed as $d_2(n_i)\hat{\sigma}$, where $\hat{\sigma}$ is an estimate of σ. If you specify a known value (σ_0) for σ, the central line indicates the value of $d_2(n_i)\sigma_0$. The central line on the range chart varies with n_i.

Control Limits

You can compute the limits

- as a specified multiple (k) of the standard errors of M_i and R_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that M_i or R_i exceeds its limits

The following table provides the formulas for the limits:

Table 43.23. Limits for Median and Range Charts

Control Limits	
Median Chart	LCL = lower limit = $\bar{M} - k\hat{\sigma}e_M(n_i)$
	UCL = upper limit = $\bar{M} + k\hat{\sigma}e_M(n_i)$
Range Chart	LCL = lower control limit = $\max(d_2(n_i)\hat{\sigma} - kd_3(n_i)\hat{\sigma}, 0)$
	UCL = upper control limit = $d_2(n_i)\hat{\sigma} + kd_3(n_i)\hat{\sigma}$

Probability Limits	
Median Chart	LCL = lower limit = $\bar{M} - Q_{\alpha/2}(n_i)\hat{\sigma}$
	UCL = upper limit = $\bar{M} + Q_{1-\alpha/2}(n_i)\hat{\sigma}$
Range Chart	LCL = lower limit = $D_{\alpha/2}\hat{\sigma}$
	UCL = upper limit = $D_{1-\alpha/2}\hat{\sigma}$

In Table 43.23, replace $\bar{M}$ with $\overline{\overline{X}}$ if you specify MEDCENTRAL=AVGMEAN, and replace $\bar{M}$ with $\tilde{M}$ if you specify MEDCENTRAL=MEDMED. Replace $\bar{M}$ with μ_0 if you specify μ_0 with the MU0= option, and replace $\hat{\sigma}$ with σ_0 if you specify σ_0 with the SIGMA0= option.

The formulas assume that the data are normally distributed. Note that the limits for both charts vary with n_i and that the probability limits for R_i are asymmetric around the central line.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify μ_0 with the MU0= option or with the variable _MEAN_ in the LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in the LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables can be saved:

Table 43.24. OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
_CP_	capability index C_p
CPK	capability index C_{pk}
CPL	capability index CPL
CPM	capability index C_{pm}
CPU	capability index CPU
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLM	lower control limit for subgroup median
LCLR	lower control limit for subgroup range
LIMITN	sample size associated with the control limits
LSL	lower specification limit
MEAN	estimate of process mean ($\bar{M}$, $\tilde{M}$, $\overline{\overline{X}}$, or μ_0)
_R_	value of central line on range chart
SIGMAS	multiple (k) of standard error of M_i or R_i
STDDEV	process standard deviation ($\hat{\sigma}$ or σ_0)

Table 43.24. (continued)

Variable	Description
SUBGRP	*subgroup-variable* specified in the MRCHART statement
TARGET	target value
TYPE	type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLM	upper control limit for subgroup median
UCLR	upper control limit for subgroup range
USL	upper specification limit
VAR	*process* specified in the XRCHART statement

Notes:

1. If the control limits vary with subgroup sample size, the special missing value *V* is assigned to the variables _LIMITN_, _LCLM_, _UCLM_, _LCLR_, _R_, and _UCLR_.
2. If the limits are defined in terms of a multiple k of the standard errors of M_i and R_i, the value of _ALPHA_ is computed as $\alpha = 2(1 - F_{med}(k, n))$, where $F_{med}(\cdot, n)$ is the cumulative distribution function of the median of a random sample of n standard normally distributed observations, and n is the value of _LIMITN_. If _LIMITN_ has the special missing value *V*, this value is assigned to _ALPHA_.
3. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = F^{-1}_{med}(1 - \alpha/2, n)$, where $F^{-1}_{med}(\cdot, n)$ is the inverse distribution function of the median of a random sample of n standard normally distributed observations, and n is the value of _LIMITN_. If _LIMITN_ has the special missing value *V*, this value is assigned to _SIGMAS_.
4. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you provide a target value with the TARGET= option. See "Capability Indices" on page 1774 for computational details.
5. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the MRCHART statement. For an example of an OUTLIMITS= data set, see "Saving Control Limits" on page 1445.

OUTHISTORY= Data Set

The OUTHISTORY= option saves subgroup summary statistics. The following variables are saved:

- the *subgroup-variable*
- a subgroup median variable named by *process* suffixed with *M*
- a subgroup range variable named by *process* suffixed with *R*
- a subgroup sample size variable named by *process* suffixed with *N*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Variables containing subgroup medians, ranges, and sample sizes are created for each *process* specified in the MRCHART statement. For example, consider the following statements:

```
proc shewhart data=steel;
   mrchart (width diameter)*lot / outhistory=summary;
run;
```

The data set SUMMARY contains variables named LOT, WIDTHM, WIDTHR, WIDTHN, DIAMTERM, DIAMTERR, and DIAMTERN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 1444.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables are saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on median chart
EXLIMR	control limit exceeded on range chart
LCLM	lower control limit for median
LCLR	lower control limit for range
LIMITN	nominal sample size associated with the control limits
MEAN	estimate of process mean ($\bar{M}$, $\tilde{M}$, $\overline{\overline{X}}$, or μ_0)
_R_	average range
SIGMAS	multiple (k) of the standard error associated with control limits
subgroup	values of the subgroup variable
SUBM	subgroup median
SUBN	subgroup sample size
SUBR	subgroup range
TESTS	tests for special causes signaled on median chart
TESTS2	tests for special causes signaled on range chart
UCLM	upper control limit for mean
UCLR	upper control limit for range
VAR	*process* specified in the MRCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The k^{th} character of a value of _TESTS_ is k if Test k is positive at that subgroup. For example, if you request all eight tests and Tests 2 and 8 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, an 8 for the eighth character, and blanks for the other six characters.
3. The variable _TESTS2_ is saved if you specify the TESTS2= option. The k^{th} character of a value of _TESTS2_ is k if Test k is positive at that subgroup.
4. The variables _EXLIM_, _EXLIMR_, _TESTS_, and _TESTS2_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example of an OUTTABLE= data set, see "Saving Control Limits" on page 1445.

ODS Tables

The following table summarizes the ODS tables that you can request with the MRCHART statement.

Table 43.25. ODS Tables Produced with the MRCHART Statement

Table Name	Description	Options
MRCHART	median and R chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the MRCHART statement must be a SAS variable in the DATA= data set. This variable provides measurements that must be grouped into subgroup samples indexed by the values of the *subgroup-variable*. The *subgroup-variable*, which is specified in the MRCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. If the t^{th} subgroup contains n_i items, there should be n_i consecutive observations for which the value of the *subgroup-variable* is the index of the t^{th} subgroup. For example, if each subgroup contains five items and there are 30 subgroup samples, the DATA= data set should contain 150 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the DATA= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 1936).

For an example of a DATA= data set, see "Creating Charts for Medians and Ranges from Raw Data" on page 1438.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set CONLIMS:*

```
proc shewhart data=info limits=conlims;
   mrchart weight*batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

*In Release 6.09 and in earlier releases, it is necessary to specify the READLIMITS option.

- the variables _LCLM_, _MEAN_, _UCLM_, _LCLR_, _R_, and _UCLR_, which specify the control limits directly
- the variables _MEAN_ and _STDDEV_, which are used to calculate the control limits according to the equations in Table 43.23 on page 1464

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are `ESTIMATE`, `STANDARD`, `STDMU`, and `STDSIGMA`.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1447.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This allows you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART procedures or to read output data sets created with SAS summarization procedures, such as PROC UNIVARIATE.

A HISTORY= data set used with the MRCHART statement must contain the following variables:

- the *subgroup-variable*
- a subgroup mean variable for each *process*
- a subgroup median variable for each *process*
- a subgroup range variable for each *process*
- a subgroup sample size variable for each *process*

The names of the subgroup mean, subgroup median, subgroup range, and subgroup sample size variables must be the *process* name concatenated with the special suffix characters X, M, R, and N, respectively. You must provide the subgroup mean variable only if you specify the MEDCENTRAL=AVGMEAN option.

For example, consider the following statements:

```
proc shewhart history=summary;
   mrchart (weight yldstren)*batch / medcentral=avgmean;
run;
```

The data set SUMMARY must include the variables BATCH, WEIGHTX, WEIGHTM, WEIGHTR, WEIGHTN, YLDSRENX, YLDSRENM, YLDSRENR, and YLDSRENN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 1936 for an example).

For an example of a HISTORY= data set, see "Creating Charts for Medians and Ranges from Summary Data" on page 1440.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure or to read data sets created by other SAS procedures. Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in Chapter 56, "Specialized Control Charts."

The following table lists the variables required in a TABLE= data set used with the MRCHART statement:

Table 43.26. Variables Required in a TABLE= Data Set

Variable	Description
LCLM	lower control limit for median
LCLR	lower control limit for range
LIMITN	nominal sample size associated with the control limits
MEAN	process mean
_R_	average range
subgroup-variable	values of the *subgroup-variable*
SUBM	subgroup median
SUBN	subgroup sample size
SUBR	subgroup range
UCLM	upper control limit for median
UCLR	upper control limit for range

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes for subgroup medians and must be a character variable of length 8.
- _TESTS2_ (if the TESTS2= option is specified). This variable is used to flag tests for special causes for subgroup ranges and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1445.

Methods for Estimating the Standard Deviation

When control limits are determined from the input data, two methods are available for estimating the process standard deviation σ.

Default Method

The default estimate for σ is

$$\hat{\sigma} = \frac{R_1/d_2(n_1) + \cdots + R_N/d_2(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, and R_i is the sample range of the observations $x_{i1}, \ldots, x_{in_i}$ in the i^{th} subgroup.

A subgroup range R_i is included in the calculation only if $n_i \geq 2$. The unbiasing factor $d_2(n_i)$ is defined so that, if the observations are normally distributed, the expected value of R_i is equal to $d_2(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the *ASTM Manual on Presentation of Data and Control Chart Analysis* (1976).

MVLUE Method

If you specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). The MVLUE is a weighted average of N unbiased estimates of σ of the form $R_i/d_2(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{f_1 R_1/d_2(n_1) + \cdots + f_N R_N/d_2(n_N)}{f_1 + \cdots + f_N}$$

where

$$f_i = \frac{[d_2(n_i)]^2}{[d_3(n_i)]^2}$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \; geq2$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

See Example 43.1 on page 1474 for illustrations of the default and MVLUE methods.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical (median chart)	DATA=	*process*
Vertical (median chart)	HISTORY=	subgroup median variable
Vertical (median chart)	TABLE=	_SUBMED_

You can specify distinct labels for the vertical axes of the median and R charts by breaking the vertical axis into two parts with a split character. Specify the split character with the SPLIT= option. The first part labels the vertical axis of the median chart, and the second part labels the vertical axis of the R chart.

For an example, see Example 43.2 on page 1478.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples

This section provides advanced examples of the MRCHART statement.

Example 43.1. Working with Unequal Subgroup Sample Sizes

See SHWMR2 in the SAS/QC Sample Library

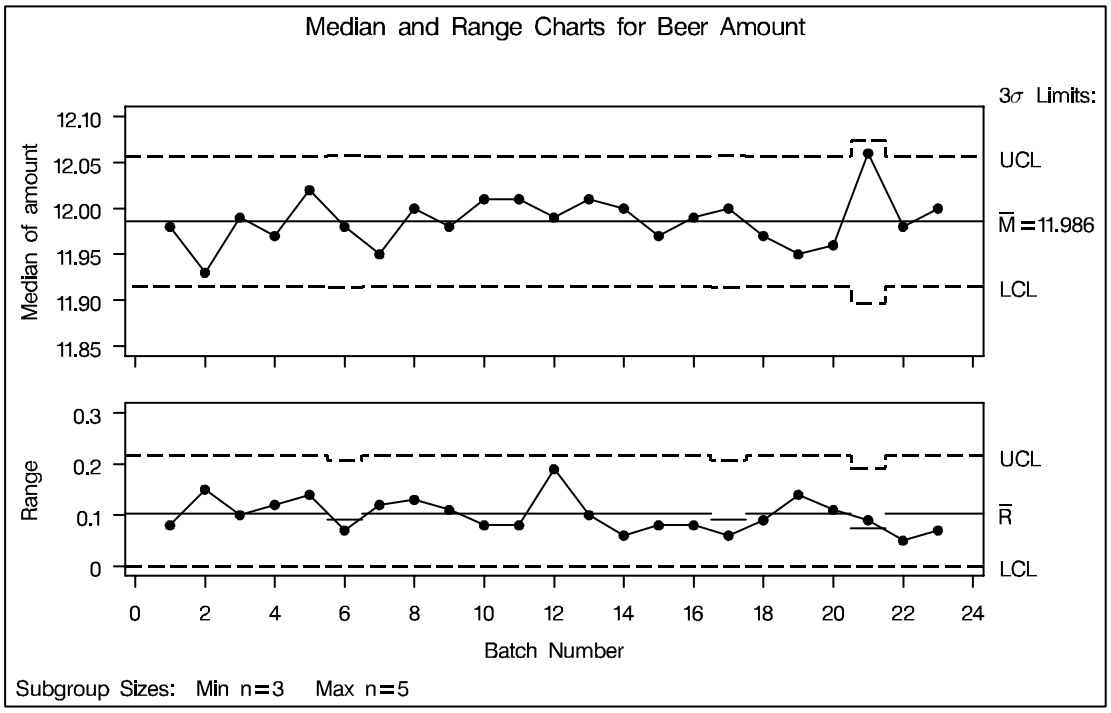
A brewery monitors its bottling process to ensure that each bottle is filled with the proper amount of beer. The following data set contains the amount of beer recorded in fluid ounces for 23 batches:

```
data beer;
   input batch size @;
   do i=1 to size;
      input amount @@;
      output;
      end;
   drop i size;
   label batch  ='Batch Number';
   datalines;
 1  5  12.01 11.97 11.93 11.98 12.00
 2  5  11.88 11.98 11.93 12.03 11.92
 3  5  11.93 11.99 12.00 12.03 11.95
 4  5  11.98 11.94 12.02 11.90 11.97
 5  5  12.02 12.02 11.98 12.04 11.90
 6  4  11.98 11.98 12.00 11.93
 7  5  11.93 11.95 12.02 11.91 12.03
 8  5  12.00 11.98 12.02 11.89 12.01
 9  5  11.98 11.93 11.99 12.02 11.91
10  5  11.97 12.02 12.05 12.01 11.97
11  5  12.02 12.01 11.97 12.02 11.94
12  5  11.93 11.83 11.99 12.02 12.01
13  5  12.01 11.98 11.94 12.04 12.01
14  5  11.98 11.96 12.02 12.00 12.00
15  5  11.97 11.99 12.03 11.95 11.96
16  5  11.99 11.95 11.96 12.03 12.01
17  4  11.99 11.97 12.03 12.01
18  5  11.94 11.96 11.98 12.03 11.97
19  5  11.97 11.87 11.90 12.01 11.95
20  5  11.96 11.94 11.96 11.98 12.05
21  3  12.06 12.07 11.98
22  5  12.01 11.98 11.96 11.97 12.00
23  5  12.00 12.02 12.03 11.99 11.96
;
run;
```

A batch is regarded as a rational subgroup. Five bottles of beer are supposed to be tested in each batch. However, in batch 6 and batch 17 only four bottles are tested, and in batch 21 only three bottles are tested. The following statements request median and range charts, shown in Output 43.1.1, for the beer amounts:

```
symbol h = .8;
title 'Median and Range Charts for Beer Amount';
proc shewhart data=beer;
   mrchart amount*batch ;
run;
```

Output 43.1.1. Median and Range Charts with Varying Sample Sizes



Since none of the subgroup medians or subgroup ranges fall outside their respective control limits, you can conclude that the process is in control.

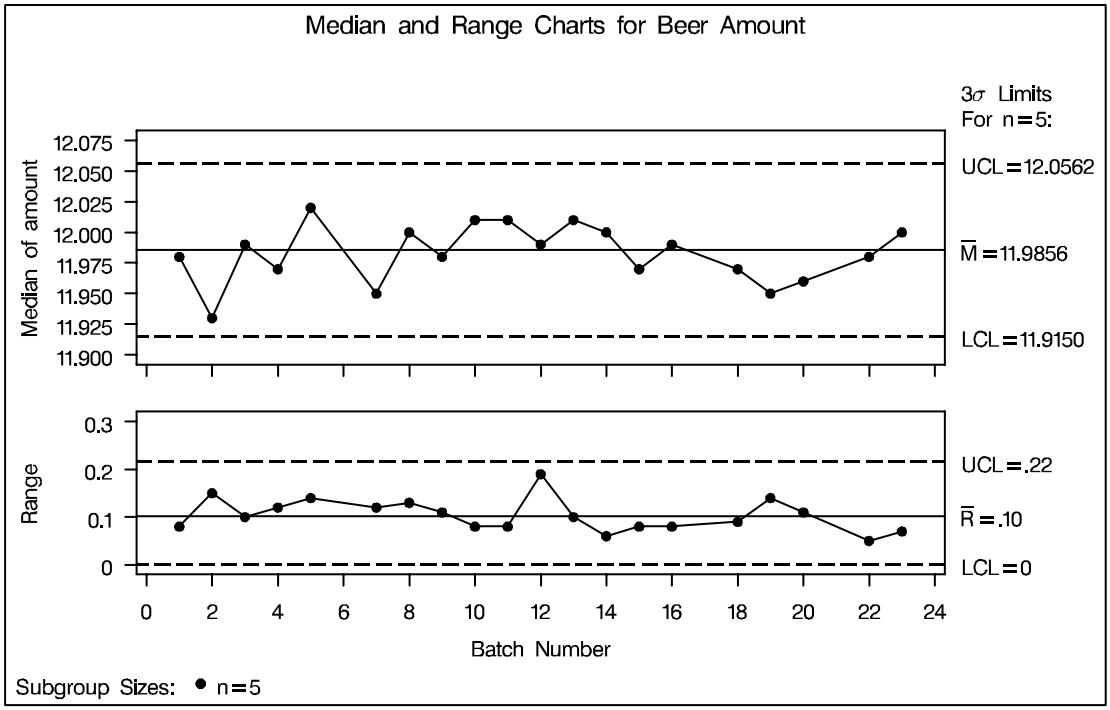
Note that the central line on the range chart and the control limits on both charts vary with the subgroup sample size. The subgroup sample size legend displays the minimum and maximum subgroup sample sizes.

The SHEWHART procedure provides various options for working with unequal subgroup sample sizes. For example, you can use the LIMITN= option to specify a fixed (nominal) sample size for the control limits, as illustrated by the following statements:

```
symbol h = .8;
title 'Median and Range Charts for Beer Amount';
proc shewhart data=beer;
   mrchart amount*batch / limitn = 5;
run;
```

The resulting charts are shown in Output 43.1.2.

Output 43.1.2. Control Limits Based on Fixed Sample Size

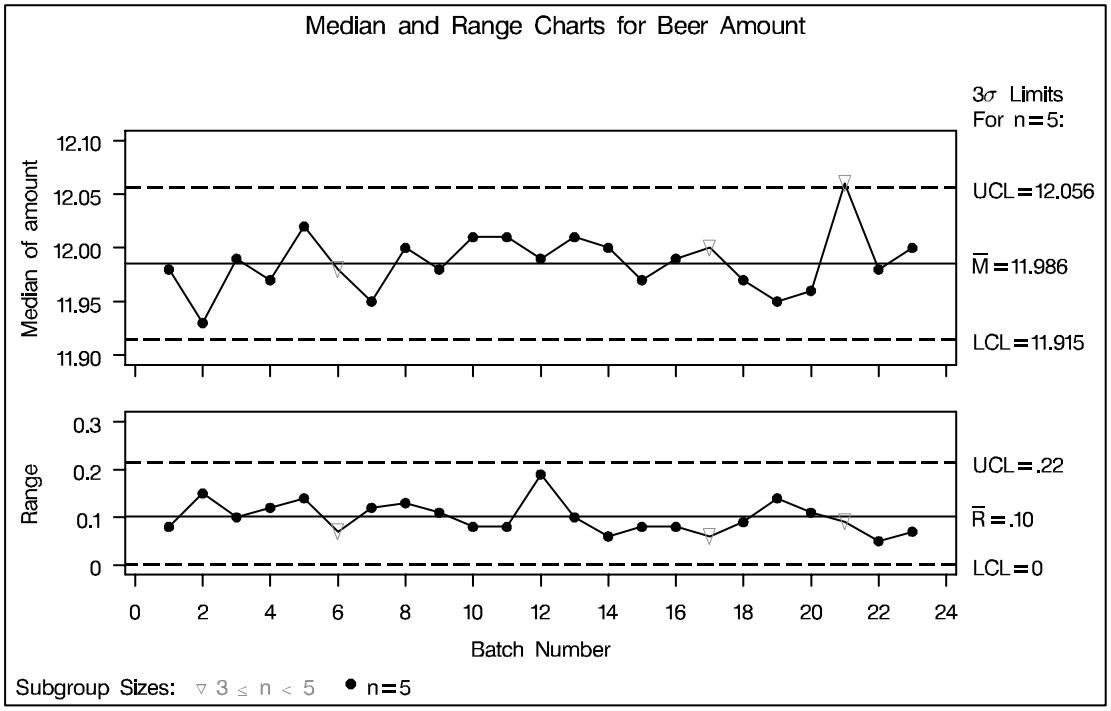


Note that the points displayed on the chart are those corresponding to subgroups whose sample size matches the nominal sample size (five) specified with the LIMITN= option. Points are not plotted for batches 6, 17, and 21. To display points for all subgroups (regardless of subgroup sample size), specify the ALLN option. The following statements produce the charts shown in Output 43.1.3:

```
symbol h = .8;
symbol2 color = rose;
title 'Median and Range Charts for Beer Amount';
proc shewhart data=beer;
   mrchart amount*batch / limitn = 5
                          alln
                          nmarkers;
run;
```

The NMARKERS option requests special symbols that identify points for which the subgroup sample size differs from the nominal sample size. In Output 43.1.3, the median amount for batch 21 exceeds the upper control limits, indicating that the process is not in control. This illustrates the approximate nature of fixed control limits used with subgroup samples of varying sizes.

Output 43.1.3. Displaying All Subgroups Regardless of Sample Size



You can use the SMETHOD= option to determine how the process standard deviation σ is to be estimated when the subgroup sample sizes vary. The default method computes σ as an unweighted average of subgroup estimates of σ. The MVLUE method assigns greater weight to estimates of σ from subgroups with larger sample sizes. If the subgroup sample sizes are constant, the MVLUE method reduces to the NOWEIGHT method.

For details, see "Methods for Estimating the Standard Deviation" on page 1472. The following statements estimate σ using both methods:

```
proc shewhart data=beer;
   mrchart amount*batch / outindex  = 'Default'
                          outlimits = blim1
                          nochart;
   mrchart amount*batch / smethod   = mvlue
                          outindex  = 'MVLUE'
                          outlimits = blim2
                          nochart;
run;

data blimits;
   set blim1 blim2;
run;
```

The estimates are saved as values of the variable _STDDEV_ in the data set BLIMITS, which is listed in Output 43.1.4. The bookkeeping variable _INDEX_ identifies the estimate.

Output 43.1.4. The Data Set BLIMITS

The Data Set BLIMITS

VAR	_SUBGRP_	_INDEX_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLM_	_MEAN_	_UCLM_	_LCLR_	_R_	_UCLR_	_STDDEV_
amount	batch	Default	ESTIMATE	V	V	3	V	11.9856	V	V	V	V	0.043938
amount	batch	MVLUE	ESTIMATE	V	V	3	V	11.9856	V	V	V	V	0.044004

In the data set BLIMITS, the variables _LIMITN_, _ALPHA_ , _LCLM_, _UCLM_, _LCLR_, _R_, and _UCLR_ have been assigned the special missing value *V*. This indicates that the quantities represented by these variables vary with the subgroup sample size.

Example 43.2. Specifying Axis Labels

See SHWMR3 in the SAS/QC Sample Library

This example illustrates various methods for specifying axis labels and other axis features for median and range charts. For further details, see "Labeling Axes" on page 1966.

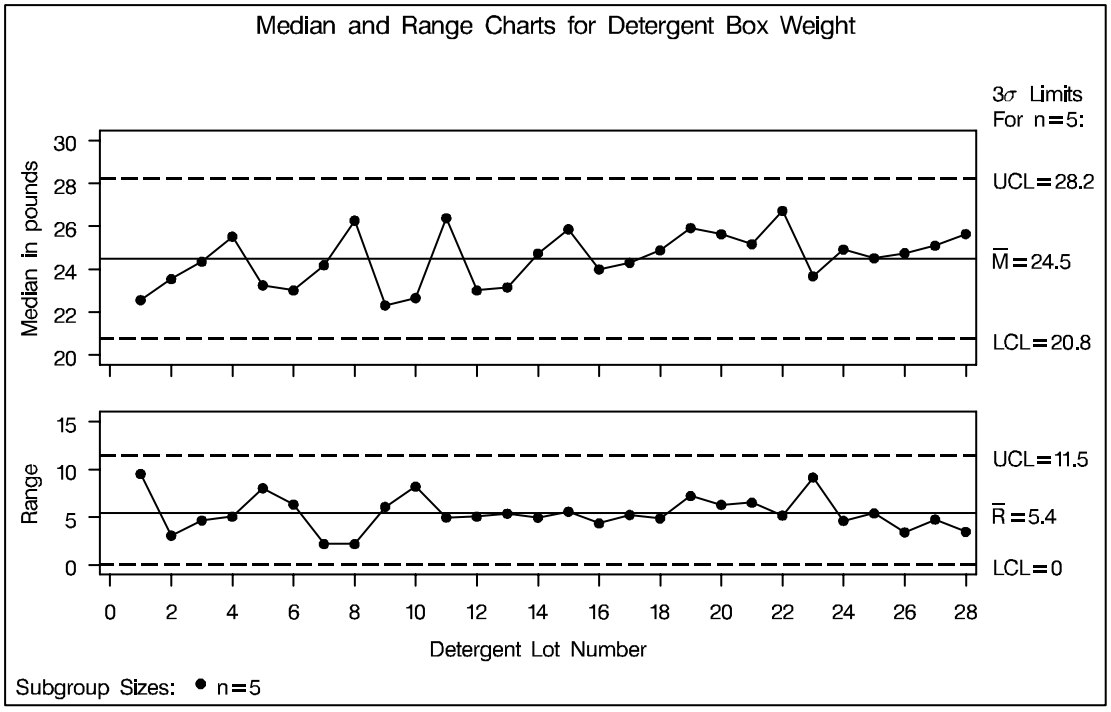
The charts in Figure 43.2 on page 1440, which are based on the data set DETERGENT introduced in the "Getting Started" section on page 1438, display default labels for the horizontal and vertical axes. You can specify axis labels by associating labels with the *process* and *subgroup* variables as illustrated by the following statements:

```
symbol h = .8;
title 'Median and Range Charts for Detergent Box Weight';
proc shewhart data=detergent;
   mrchart weight*lot / split   = '/';
   label lot     = 'Detergent Lot Number'
         weight = 'Median in pounds/Range';
run;
```

The charts are shown in Output 43.2.1. The horizontal axis label is the label associated with the *subgroup-variable* LOT. The vertical axis label for the median chart, referred to as the primary vertical axis label, is the first portion of the label associated with the *process* variable WEIGHT, up to but not including the split character, which is specified with the SPLIT= option. The vertical axis label for the range chart,

referred to as the secondary vertical axis label, is the second portion of the label associated with WEIGHT.

Output 43.2.1. Customized Axis Labels Using Variable Labels



When the input data set is a HISTORY= data set, the vertical axis labels are determined by the label associated with the subgroup median variable. This is illustrated by the following statements, which use the data set DETSUM introduced on page 1441:

```
title 'Median and Range Charts for Detergent Box Weight';
symbol v=dot;
proc shewhart history=detsum;
   mrchart weight*lot / split = '/';
   label lot     = 'Detergent Lot Number'
         weightm = 'Median (pounds)/Range';
run;
```

The charts are identical to those in Output 43.2.1.

When the input data set is a TABLE= data set, the vertical axis labels are determined by the label associated with the subgroup median variable _SUBMED_. This is illustrated by the following statements, which use the data set DTABLE introduced in Figure 43.7 on page 1446:

```
title 'Median and Range Charts for Detergent Box Weight';
symbol v=dot;
proc shewhart table=dtable;
   mrchart weight*lot / split = '/';
   label lot      = 'Detergent Lot Number'
         _submed_ = 'Median (pounds)/Range';
run;
```

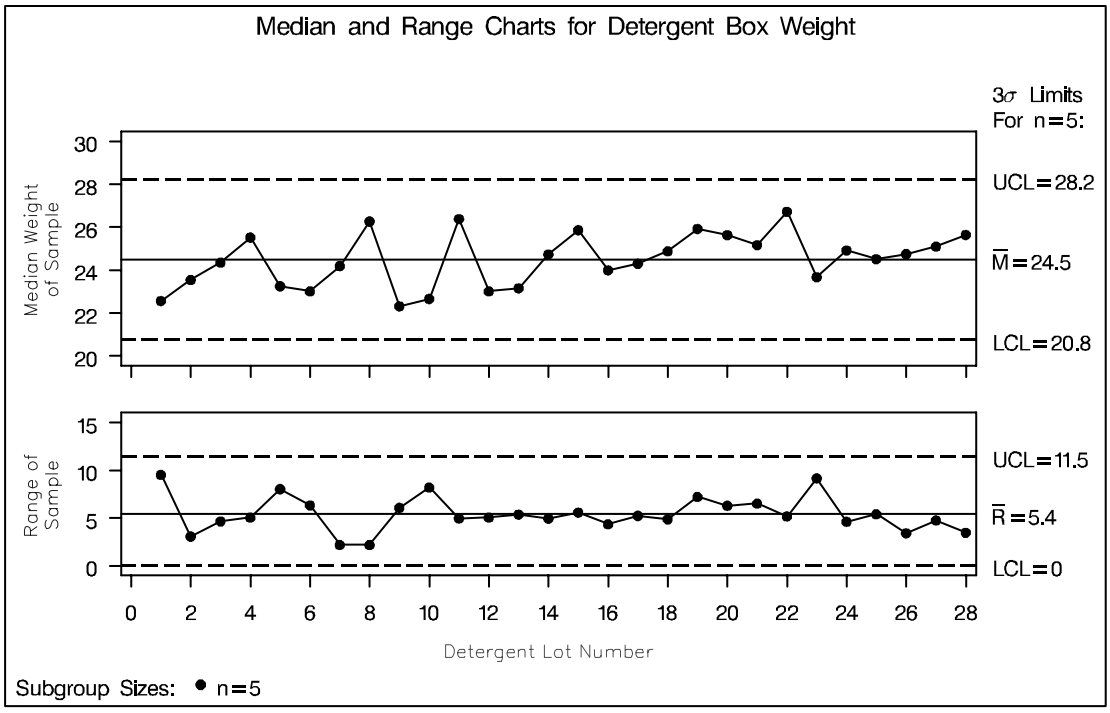
The charts are identical to those in Output 43.2.1.

When you are creating charts on graphics devices, you can use AXIS statements to enhance the appearance of the axes. This method is illustrated by the following statements:

```
symbol h = .8;
title 'Median and Range Charts for Detergent Box Weight';
proc shewhart data=detergent;
   mrchart weight*lot / haxis   = axis1
                        vaxis   = axis2
                        vaxis2  = axis3;
axis1 label=(c=red  f=simplex 'Detergent Lot Number'          );
axis2 label=(c=blue f=simplex 'Median Weight' j=c 'of Sample' );
axis3 label=(c=blue f=simplex 'Range of'      j=c 'Sample'    );
run;
```

The charts are shown in Output 43.2.2.

Output 43.2.2. Customized Axis Labels Using AXIS Statements



You can use AXIS statements to customize a variety of axis features. For details, see *SAS/GRAPH Software: Reference*.

Chapter 44
NPCHART Statement

Chapter Contents

OVERVIEW 1483

GETTING STARTED 1484
- Creating np Charts from Count Data 1484
- Creating np Charts from Summary Data 1486
- Saving Proportions of Nonconforming Items 1488
- Saving Control Limits 1489
- Reading Preestablished Control Limits 1492

SYNTAX 1494
- Summary of Options 1495

DETAILS 1506
- Constructing Charts for Number Nonconforming (np Charts) 1506
- Output Data Sets 1508
- ODS Tables 1511
- Input Data Sets 1511
- Axis Labels 1514
- Missing Values 1514

EXAMPLES 1515
- Example 44.1. Applying Tests for Special Causes 1515
- Example 44.2. Specifying Standard Average Proportion 1517
- Example 44.3. Working with Unequal Subgroup Sample Sizes 1518
- Example 44.4. Specifying Control Limit Information 1521

Chapter 44
NPCHART Statement

Overview

The NPCHART statement creates np charts for the numbers of nonconforming (defective) items in subgroup samples.

You can use options in the NPCHART statement to

- compute control limits from the data based on a multiple of the standard error of the numbers of nonconforming items or as probability limits
- tabulate subgroup sample sizes, numbers of nonconforming items, control limits, and other information
- save control limits in an output data set
- save subgroup sample sizes and proportions of nonconforming items in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify a known (standard) proportion of nonconforming items for computing control limits
- specify the data as counts, proportions, or percentages of nonconforming items
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

Getting Started

This section introduces the NPCHART statement with simple examples that illustrate commonly used options. Complete syntax for the NPCHART statement is presented in the "Syntax" section on page 1494, and advanced examples are given in the "Examples" section on page 1515.

Creating np Charts from Count Data

See SHWNP1 in the SAS/QC Sample Library

An electronics company manufactures circuits in batches of 500 and uses an np chart to monitor the number of failing circuits. Thirty batches are examined, and the failures in each batch are counted. The following statements create a SAS data set named CIRCUITS,* which contains the failure counts:

```
data circuits;
  input batch fail @@;
  datalines;
1    5   2   6   3  11   4   6    5   4
6    9   7  17   8  10   9  12   10   9
11   8  12   7  13   7  14  15   15   8
16  18  17  12  18  16  19   4   20   7
21  17  22  12  23   8  24   7   25  15
26   6  27   8  28  12  29   7   30   9
;
run;
```

A partial listing of CIRCUITS is shown in Figure 44.1.

Number of Failing Circuits

batch	fail
1	5
2	6
3	11
4	6
5	4
.	.
.	.
.	.

Figure 44.1. The Data Set CIRCUITS

There is a single observation for each batch. The variable BATCH identifies the subgroup sample and is referred to as the *subgroup-variable*. The variable FAIL contains the number of nonconforming items in each subgroup sample and is referred to as the *process variable* (or *process* for short).

*This data set is also used in the "Getting Started" section of Chapter 45, "PCHART Statement."

The following statements create the np chart shown in Figure 44.2:

```
symbol h = .8;
title 'np Chart for the Number of Failing Circuits';
proc shewhart data=circuits;
   npchart fail*batch / subgroupn = 500;
run;
```

This example illustrates the basic form of the NPCHART statement. After the keyword NPCHART, you specify the *process* to analyze (in this case, FAIL), followed by an asterisk and the *subgroup-variable* (BATCH).

The input data set is specified with the DATA= option in the PROC SHEWHART statement. The SUBGROUPN= option specifies the number of items in each subgroup sample and is required with a DATA= input data set. The SUBGROUPN= option specifies one of the following:

- a constant subgroup sample size (in this case)
- a variable in the input data set whose values provide the subgroup sample sizes (see the next example)

Options such as SUBGROUPN= are specified after the slash (/) in the NPCHART statement. A complete list of options is presented in the "Syntax" section on page 1494.

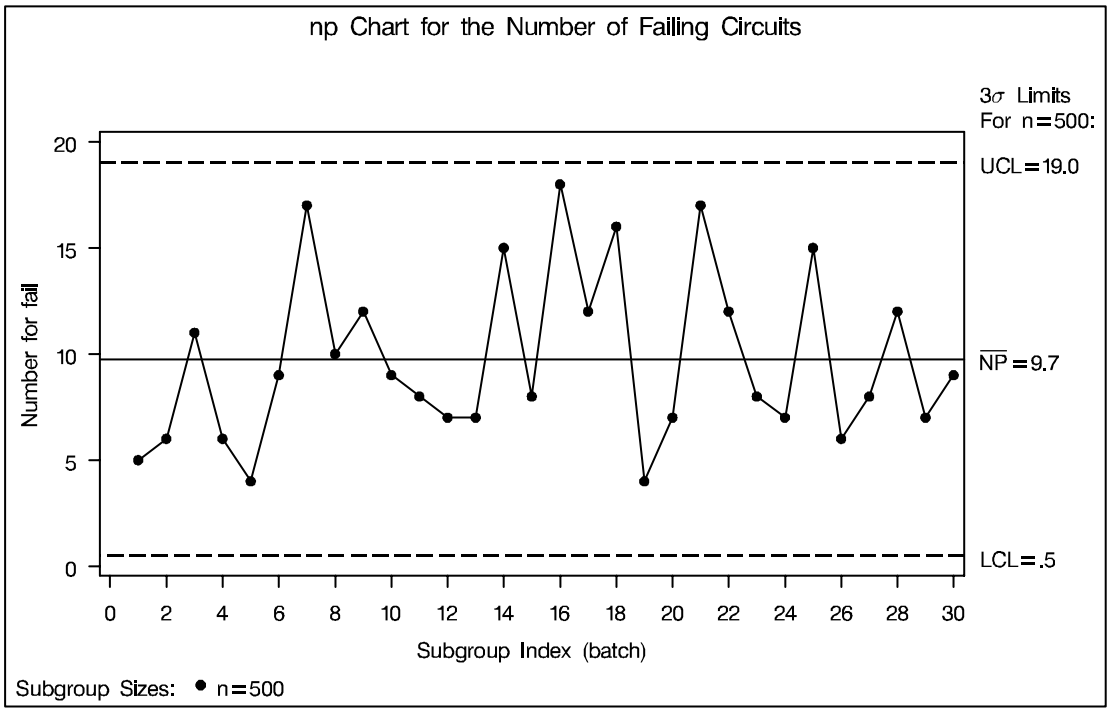


Figure 44.2. An np Chart for Circuit Failures

Each point on the np chart represents the number of nonconforming items for a particular subgroup. For instance, the value plotted for the first batch is 5.

Since all the points fall within the control limits, it can be concluded that the process is in statistical control.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in "Control Limits" on page 1507. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1492. For computational details, see "Constructing Charts for Number Nonconforming (np Charts)" on page 1506. For more details on reading raw data, see "DATA= Data Set" on page 1511.

Creating np Charts from Summary Data

See SHWNP1 in the SAS/QC Sample Library

The previous example illustrates how you can create np charts using raw data (counts of nonconforming items). However, in many applications, the data are provided in summarized form as proportions or percentages of nonconforming items. This example illustrates how you can use the NPCHART statement with data of this type.

The following data set provides the data from the preceding example in summarized form:

```
data cirprop;
   input batch pfailed @@;
   sizes=500;
datalines;
 1  0.010   2  0.012   3  0.022   4  0.012   5  0.008
 6  0.018   7  0.034   8  0.020   9  0.024  10  0.018
11  0.016  12  0.014  13  0.014  14  0.030  15  0.016
16  0.036  17  0.024  18  0.032  19  0.008  20  0.014
21  0.034  22  0.024  23  0.016  24  0.014  25  0.030
26  0.012  27  0.016  28  0.024  29  0.014  30  0.018
;
run;
```

A partial listing of CIRPROP is shown in Figure 44.3. The subgroups are still indexed by BATCH. The variable PFAILED contains the proportions of nonconforming items, and the variable SAMPSIZE contains the subgroup sample sizes.

Subgroup Proportions of Nonconforming Items

batch	pfailed	sizes
1	0.010	500
2	0.012	500
3	0.022	500
.	.	.
.	.	.
.	.	.

Figure 44.3. The Data Set CIRPROP

The following statements create an np chart identical to the one in Figure 44.2:

```
title 'np Chart for the Number of Failing Circuits';
symbol v=dot;
proc shewhart data=cirprop;
   npchart pfailed*batch / subgroupn=sampsize
                           dataunit =proportion;
label pfailed = 'Number of FAIL';
run;
```

The DATAUNIT= option specifies that the values of the *process* (PFAILED) are proportions of nonconforming items. By default, the values of the *process* are assumed to be counts of nonconforming items (see the previous example).

Alternatively, you can read the data set CIRPROP by specifying it as a HISTORY= data set in the PROC SHEWHART statement. A HISTORY= data set used with the NPCHART statement must contain the following variables:

- subgroup variable
- subgroup proportion of nonconforming items variable
- subgroup sample size variable

Furthermore, the names of the subgroup proportion and sample size variables must begin with the *process* name specified in the NPCHART statement and end with the special suffix characters P and N, respectively.

To specify CIRPROP as a HISTORY= data set and FAIL as the *process*, you must rename the variables PFAILED and SAMPSIZE to FAILP and FAILN, respectively. The following statements temporarily rename PFAILED and SAMPSIZE for the duration of the procedure step:

```
title 'np Chart for the Number of Failing Circuits';
proc shewhart history=cirprop(rename=(pfailed =failp
                                      sizes=failn )) lineprinter;
   npchart fail*batch='*';
run;
```

The resulting np chart is shown in Figure 44.4. Since the LINEPRINTER option is specified in the PROC SHEWHART statement, line printer output is produced. The asterisk specified in single quotes after the *subgroup-variable* indicates the character used to plot points. This character must follow an equal sign.

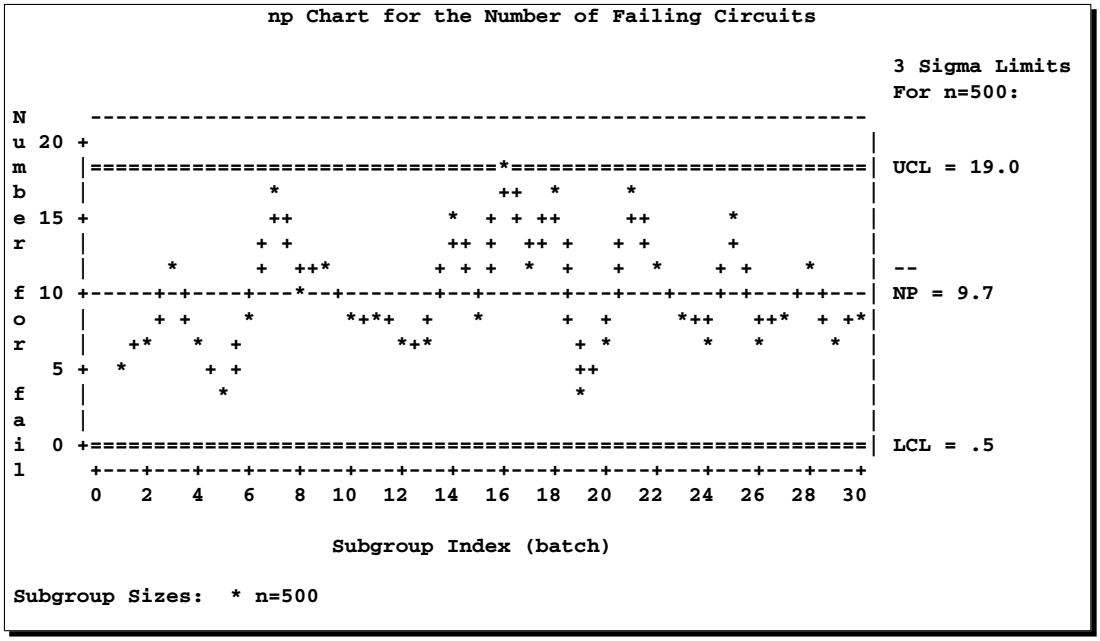


Figure 44.4. An np Chart for Circuit Failures

In this example, it is more convenient to use CIRPROP as a DATA= data set than as a HISTORY= data set. As illustrated in the next example, it is generally more convenient to use the HISTORY= option for input data sets that have been created previously by the SHEWHART procedure as OUTHISTORY= data sets.

For more information, see "HISTORY= Data Set" on page 1512.

Saving Proportions of Nonconforming Items

See SHWNP1 in the SAS/QC Sample Library

In this example, the NPCHART statement is used to create a data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read the number of nonconforming items from the data set CIRCUITS (see page 1484) and create a summary data set named CIRHIST:

```
proc shewhart data=circuits;
   npchart fail*batch / subgroupn  = 500
                        outhistory = cirhist
                        nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 44.2. Figure 44.5 contains a partial listing of CIRHIST.

Subgroup Proportions of Failing Circuits

batch	failP	fail N
1	0.010	500
2	0.012	500
3	0.022	500
4	0.012	500
5	0.008	500
.	.	.
.	.	.
.	.	.

Figure 44.5. The Data Set CIRHIST

There are three variables in the data set CIRHIST.

- BATCH contains the subgroup index.
- FAILP contains the subgroup proportion of nonconforming items.
- FAILN contains the subgroup sample size.

Note that the variables containing the subgroup proportions of nonconforming items and subgroup sample sizes are named by adding the suffix characters *P* and *N* to the *process* FAIL specified in the NPCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 1509.

Saving Control Limits

See SHWNP1 in the SAS/QC Sample Library

You can save the control limits for an *np* chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1492) or modify the limits with a DATA step program.

The following statements read the number of nonconforming items per subgroup from the data set CIRCUITS (see page 1484) and save the control limits displayed in Figure 44.2 in a data set named CIRLIM:

```
proc shewhart data=circuits;
   npchart fail*batch / subgroupn=500
                        outlimits=cirlim
                        nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. The data set CIRLIM is listed in Figure 44.6.

Control Limits for the Number of Failing Circuits

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_NP_	_SIGMAS_	_P_	_LCLNP_	_UCLNP_	_UCLNP_
fail	batch	ESTIMATE	500	.005040334	3	0.019467	0.46539	9.73333	19.0013

Figure 44.6. The Data Set CIRLIM Containing Control Limit Information

The data set CIRLIM contains one observation with the limits for *process* FAIL. The variables _LCLNP_ and _UCLNP_ contain the lower and upper control limits, and the variable _NP_ contains the central line. The variable _P_ contains the average proportion of nonconforming items. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the value of _P_ is an estimate or a standard value.

For more information, see "OUTLIMITS= Data Set" on page 1508.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=circuits;
   npchart fail*batch / subgroupn=500
                        outtable=cirtable
                        nochart;
run;
```

The data set CIRTABLE is listed in Figure 44.7.

This data set contains one observation for each subgroup sample. The variables _SUBNP_ and _SUBN_ contain the subgroup numbers of nonconforming items and subgroup sample sizes, respectively. The variables _LCLNP_ and _UCLNP_ contain the lower and upper control limits, and the variable _NP_ contains the central line. The variables _VAR_ and BATCH contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1509.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read CIRTABLE and display an np chart (not shown here) identical to the chart in Figure 44.2:

```
title 'np Chart for the Number of Failing Circuits';
proc shewhart table=cirtable;
   npchart fail*batch;
run;
```

Number Nonconforming and Control Limit Information

VAR	batch	_SIGMAS_	_LIMITN_	_SUBN_	_LCLNP_	_SUBNP_	_NP_	_UCLNP_	_EXLIM_
fail	1	3	500	500	0.46539	5	9.73333	19.0013	
fail	2	3	500	500	0.46539	6	9.73333	19.0013	
fail	3	3	500	500	0.46539	11	9.73333	19.0013	
fail	4	3	500	500	0.46539	6	9.73333	19.0013	
fail	5	3	500	500	0.46539	4	9.73333	19.0013	
fail	6	3	500	500	0.46539	9	9.73333	19.0013	
fail	7	3	500	500	0.46539	17	9.73333	19.0013	
fail	8	3	500	500	0.46539	10	9.73333	19.0013	
fail	9	3	500	500	0.46539	12	9.73333	19.0013	
fail	10	3	500	500	0.46539	9	9.73333	19.0013	
fail	11	3	500	500	0.46539	8	9.73333	19.0013	
fail	12	3	500	500	0.46539	7	9.73333	19.0013	
fail	13	3	500	500	0.46539	7	9.73333	19.0013	
fail	14	3	500	500	0.46539	15	9.73333	19.0013	
fail	15	3	500	500	0.46539	8	9.73333	19.0013	
fail	16	3	500	500	0.46539	18	9.73333	19.0013	
fail	17	3	500	500	0.46539	12	9.73333	19.0013	
fail	18	3	500	500	0.46539	16	9.73333	19.0013	
fail	19	3	500	500	0.46539	4	9.73333	19.0013	
fail	20	3	500	500	0.46539	7	9.73333	19.0013	
fail	21	3	500	500	0.46539	17	9.73333	19.0013	
fail	22	3	500	500	0.46539	12	9.73333	19.0013	
fail	23	3	500	500	0.46539	8	9.73333	19.0013	
fail	24	3	500	500	0.46539	7	9.73333	19.0013	
fail	25	3	500	500	0.46539	15	9.73333	19.0013	
fail	26	3	500	500	0.46539	6	9.73333	19.0013	
fail	27	3	500	500	0.46539	8	9.73333	19.0013	
fail	28	3	500	500	0.46539	12	9.73333	19.0013	
fail	29	3	500	500	0.46539	7	9.73333	19.0013	
fail	30	3	500	500	0.46539	9	9.73333	19.0013	

Figure 44.7. The Data Set CIRTABLE

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see Chapter 56, "Specialized Control Charts,"). For more information, see "TABLE= Data Set" on page 1513.

Reading Preestablished Control Limits

See SHWNP1 in the SAS/QC Sample Library

In the previous example, the OUTLIMITS= data set CIRLIM saved control limits computed from the data in CIRCUITS. This example shows how these limits can be applied to new data provided in the following data set:

```
data circuit2;
   input batch fail;
datalines;
31  12   32   9   33  16   34   9
35   3   36   8   37  20   38   4
39   8   40   6   41  12   42  16
43   9   44   2   45  10   46   8
47  14   48  10   49  11   50   9
;
run;
```

The following statements create an np chart for the data in CIRCUIT2 using the control limits in CIRLIM:

```
symbol h = .8;
title 'np Chart for the Proportion of Failing Circuits';
proc shewhart data=circuit2 limits=cirlim;
   npchart fail*batch / subgroupn = 500;
run;
```

The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default,* this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name FAIL
- the value of _SUBGRP_ matches the *subgroup-variable* name BATCH

The resulting np chart is shown in Figure 44.8.

*In Release 6.09 and in earlier releases, it is also necessary to specify the READLIMITS option to read control limits from a LIMITS= data set.

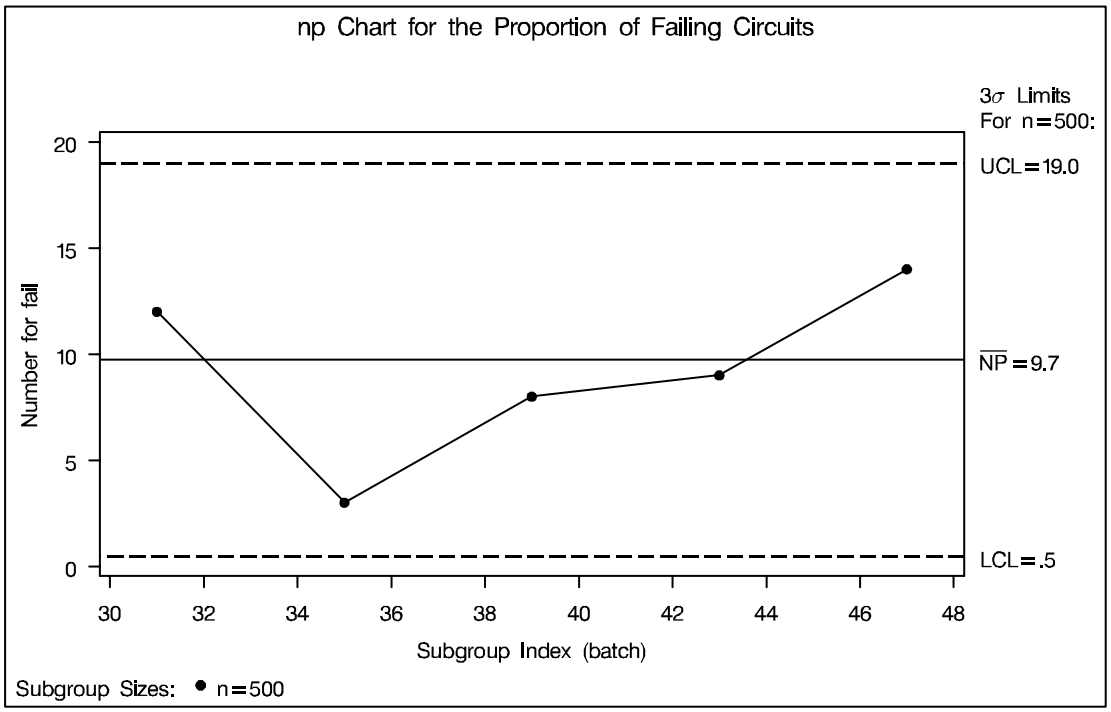


Figure 44.8. An np Chart for Second Set of Circuit Failures

The number of nonconforming items in the 37^{th} batch exceeds the upper control limit, signaling that the process is out of control.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step; see Example 44.4 on page 1521 for an example. See "LIMITS= Data Set" on page 1512 for details concerning the variables that you must provide.

Syntax

The basic syntax for the NPCHART statement is as follows:

NPCHART *process***subgroup-variable* **;**

The general form of this syntax is as follows:

NPCHART (*processes***)****subgroup-variable* <**(***block-variables* **)** >
< =*symbol-variable* | =’*character*’ > < **/** *options* >**;**

You can use any number of NPCHART statements in the SHEWHART procedure. The components of the NPCHART statement are described as follows.

process
processes
identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If numbers of nonconforming items are read from a DATA= data set, *process* must be the name of the variable containing the numbers. For an example, see “Creating np Charts from Count Data” on page 1484.
- If proportions of nonconforming items are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see “Creating np Charts from Summary Data” on page 1486.
- If numbers of nonconforming items and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see “Saving Control Limits” on page 1489.

A *process* is required. If you specify more than one process, enclose the list in parentheses. For example, the following statements request distinct np charts for REJECTS and REWORKS:

```
proc shewhart data=measures;
   npchart (rejects reworks)*sample / subgroupn=100;
run;
```

Note that when data are read from a DATA= data set, the SUBGROUPN= option, which specifies subgroup sample sizes, is required.

subgroup-variable
is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding NPCHART statement, SAMPLE is the subgroup variable. For details, see “Subgroup Variables” on page 1771.

block-variables
are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in

the legend. See "Displaying Stratification in Blocks of Observations" on page 1932 for an example.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker or character used to plot numbers of nonconforming items.

- If you produce a chart on a line printer, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce a chart on a graphics device, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOLn statements. See "Displaying Stratification in Levels of a Classification Variable" on page 1931 for an example.

character
specifies a plotting character for charts produced on line printers. For example, the following statements create an np chart using an asterisk (*) to plot the points:

```
proc shewhart data=values;
   npchart rejects*day='*' / subgroupn=100;
run;
```

options
enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The "Summary of Options" section, which follows, lists all options by function. Chapter 53, "Dictionary of Options," describes each option in detail.

Summary of Options

The following tables list the NPCHART statement options by function. For complete descriptions, see Chapter 53, "Dictionary of Options."

Table 44.1. Tabulation Options

TABLE	creates a basic table of subgroup sample sizes, subgroup numbers of nonconforming items, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with a column indicating which tests for special causes are positive

Note that specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.

Table 44.2. Plot Layout Options

ALLN	plots total number of nonconforming items for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts for a *process* only when exceptions occur
INTERVAL=*keyword*	specifies natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=*n*	specifies maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=*n*	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=*n*	specifies number of pages or screens to be used to display chart
ZEROSTD	displays np chart regardless of whether $\hat{\sigma} = 0$

Table 44.3. Reference Line Options

CHREF=*color*	specifies color for lines requested by the HREF= option
CVREF=*color*	specifies color for lines requested by the VREF= option
HREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis
HREFCHAR='*character*'	specifies line character for HREF= lines
HREFDATA= *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis
HREFLABELS= *'label1'...'labeln'*	specifies labels for HREF= lines
HREFLABPOS=*n*	specifies position of HREFLABELS= labels
LHREF=*linetype*	specifies line type for HREF= lines
LVREF=*linetype*	specifies line type for VREF= lines
NOBYREF	specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to vertical axis
VREFCHAR='*character*'	specifies line character for VREF= lines
VREFLABELS= *'label1'...'labeln'*	specifies labels for VREF= lines
VREFLABPOS=*n*	specifies position of VREFLABELS= labels

Table 44.4. Options for Specifying Tests for Special Causes

NO3SIGMACHECK	allows tests to be applied with control limits other than 3σ limits
TESTS=*value-list*| *customized-pattern-list*	specifies tests for special causes
TEST2RUN=*n*	specifies length of pattern for Test 2
TEST3RUN=*n*	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL='*label*'| (*variable*)|*keyword*	provides labels for points where test is positive
TESTLABEL*n*='*label*'	specifies label for n^{th} test for special causes
TESTNMETHOD= STANDARDIZE	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
TESTRESET=*variable*	allows tests for special causes to be reset
ZONELABELS	adds labels A, B, and C to zone lines
ZONES	adds lines delineating zones A, B, and C
ZONEVALPOS=*n*	specifies position of ZONEVALUES labels
ZONEVALUES	labels zone lines with their values

Table 44.5. Graphical Options for Displaying Tests for Special Causes

CTESTLABBOX=*color*	specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=*color*| *test-color-list*	specifies color for labels indicating points where test is positive
CTESTSYMBOL=*color*	specifies color for symbol used to plot points where test is positive
CZONES=*color*	specifies color for lines and labels delineating zones A, B, and C
LTESTS=*linetype*	specifies type of line connecting points where test is positive
LZONES=*linetype*	specifies line type for lines delineating zones A, B, and C
TESTFONT=*font*	specifies software font for labels at points where test is positive
TESTHEIGHT=*value*	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=*symbol*	specifies plot symbol for points where test is positive
TESTSYMBOLHT=*value*	specifies symbol height for points where test is positive
WTESTS=*n*	specifies width of line connecting points where test is positive

Table 44.6. Line Printer Options for Displaying Tests for Special Causes

TESTCHAR='*character*'	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
ZONECHAR='*character*'	specifies character for lines that delineate zones for tests for special causes

Table 44.7. Clipping Options

CCLIP=*color*	specifies color for plot symbol for clipped points
CLIPCHAR='*character*'	specifies plot character for clipped points
CLIPFACTOR=*value*	determines extent to which extreme points are clipped
CLIPLEGEND='*string*'	specifies text for clipping legend
CLIPLEGPOS=*keyword*	specifies position of clipping legend
CLIPSUBCHAR= '*character*'	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=*symbol*	specifies plot symbol for clipped points
CLIPSYMBOLHT=*value*	specifies symbol marker height for clipped points

Table 44.8. Block Variable Legend Options

BLOCKLABELPOS=*keyword*	specifies position of label for *block-variable* legend
BLOCKLABTYPE=*n*|*keyword*	specifies text size of *block-variable* legend
BLOCKPOS=*n*	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=*color-list*	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=*variable*|(*variables*)	specifies one or more variables whose values are colors for filling background of *block-variable* legend

Table 44.9. Axis and Axis Label Options

CAXIS=*color*	specifies color for axis lines and tick marks
CFRAME=*color*|(*color-list*)	specifies fill colors for frame for plot area
CTEXT=*color*	specifies color for tick mark values and axis labels
HAXIS=*values*|AXIS*n*	specifies major tick mark values for horizontal axis
HEIGHT=*value*	specifies height of axis label and axis legend text
HMINOR=*n*	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=*value*	specifies length of offset at both ends of horizontal axis
INTSTART=*value*	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	suppresses vertical axis truncation at zero applied by default
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=*n*	specifies thinning factor for tick mark labels on horizontal axis
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=*values*|AXIS*n*	specifies major tick mark values for vertical axis
VFORMAT=*format*	specifies format for vertical axis tick mark labels
VMINOR=*n*	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=*value*	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis
WAXIS=*n*	specifies width of axis lines

Table 44.10. Grid Options

CGRID=*color*	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=*linetype*	specifies line type for grid requested with the ENDGRID option
LGRID=*linetype*	specifies line type for grid requested with the GRID option
WGRID=*n*	specifies width of grid lines

Table 44.11. Options for Specifying Control Limits

ALPHA=*value*	requests probability limits for control charts
LIMITN=*n*|VARYING	specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	computes control limits for each *process* from the data rather than from a LIMITS= data set (Release 6.10 and later releases)
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEXES=ALL| '*label1*'...'*labeln*'	reads multiple sets of control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (Release 6.09 and earlier releases)
SIGMAS=*k*	specifies width of control limits in terms of multiple k of standard error of plotted number of nonconforming items

Table 44.12. Options for Displaying Control Limits

CINFILL=*color*	specifies color for area inside control limits
CLIMITS=*color*	specifies color of control limits, central line, and related labels
LCLLABEL='*label*'	specifies label for lower control limit
LIMLABSUBCHAR= '*character*'	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=*linetype*	specifies line type for control limits
NDECIMAL=*n*	specifies number of digits to right of decimal place in default labels for control limits and central line
NOCTL	suppresses display of central line
NOLCL	suppresses display of lower control limit
NOLIMITLABEL	suppresses labels for control limits and central line
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOLIMIT0	suppresses display of lower control limit if it is 0
NOLIMIT1	suppresses display of upper control limit if it is equal to subgroup sample size
NOUCL	suppresses display of upper control limit
NPSYMBOL='*string*'| *keyword*	specifies label for central line
UCLLABEL='*string*'	specifies label for upper control limit
WLIMITS=*n*	specifies width for control limits and central line

Table 44.13. Phase Options

CPHASELEG=*color*	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE='*string*'	specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=*value*| *keyword*	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES= ALL| '*label1*'...'*labeln*'	specifies *phases* to be read from an input data set

Table 44.14. Standard Value Options

P0=*value*	specifies known (standard) value p_0 for proportion of nonconforming items
TYPE=*keyword*	identifies whether parameters are estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set

Table 44.15. Options for Interactive Control Charts

HTML=(*variable*)	specifies a variable whose values are URLs to be associated with subgroups
HTML_LEGEND=(*variable*)	specifies a variable whose values are URLs to be associated with symbols in the symbol legend
WEBOUT=*SAS-data-set*	creates an OUTTABLE= data set with additional graphics coordinate data

Table 44.16. Input Data Set Options

DATAUNIT=*keyword*	specifies that input values are proportions or percentages (rather than counts) of nonconforming items
MISSBREAK	specifies that observations with missing values are not to be processed
SUBGROUPN=*n*|*variable*	specifies subgroup sample sizes as constant number n or as values of *variable* in a DATA= data set

Table 44.17. Output Data Set Options

OUTHISTORY=*SAS-data-set*	creates output data set containing subgroup proportions of nonconforming items and subgroup sample sizes
OUTINDEX='*string*'	specifies value of the variable _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=*SAS-data-set*	creates output data set containing control limits
OUTTABLE=*SAS-data-set*	creates output data set containing subgroup numbers of nonconforming items, subgroup sample sizes, and control limits

Table 44.18. Options for Plotting and Labeling Points

ALLLABEL=VALUE|(*variable*)	labels every point
CLABEL=*color*	specifies color for labels
CCONNECT=*color*	specifies color for line segments that connect points on chart
CFRAMELAB=*color*	specifies fill color for frame around labeled points
CNEEDLES=*color*	specifies color for needles that connect points to central line
CONNECTCHAR='*character*'	specifies character used to form line segments that connect points on chart
COUT=*color*	specifies color for portions of line segments that connect points outside control limits
COUTFILL=*color*	specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=*angle*	specifies angle at which labels are drawn
LABELFONT=*font*	specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=*value*	specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on chart
OUTLABEL=VALUE|(*variable*)	labels points outside control limits
SYMBOLCHARS='*characters*'	specifies characters indicating *symbol-variable*
SYMBOLLEGEND=NONE|*name*	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=*keyword*	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=*n*	specifies width of needles

Table 44.19. Graphical Enhancement Options

ANNOTATE=*SAS-data-set*	specifies annotate data set that adds features to chart
DESCRIPTION='*string*'	specifies string that appears in the description field of the PROC GREPLAY master menu
FONT=*font*	specifies software font for labels and legends on charts
NAME='*string*'	specifies name that appears in the name field of the PROC GREPLAY master menu
PAGENUM='*string*'	specifies the form of the label used in pagination
PAGENUMPOS=*keyword*	specifies the position of the page number requested with the PAGENUM= option

Table 44.20. Star Options

CSTARCIRCLES=*color*	specifies color for circles specified by the STARCIRCLES= option
CSTARFILL=*color*| *(variable)*	specifies color for filling stars
CSTAROUT=*color*	specifies outline color for stars exceeding inner or outer circles
CSTARS=*color*| *(variable)*	specifies color for outlines of stars
LSTARCIRCLES= *linetypes*	specifies line types for STARCIRCLES= circles
LSTARS=*linetype*| *(variable)*	specifies line types for outlines of stars requested with the STARVERTICES= option
STARBDRADIUS=*value*	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=*value-list*	specifies reference circles for stars
STARINRADIUS=*value*	specifies inner radius of stars
STARLABEL=*keyword*	specifies vertices to be labeled
STARLEGEND=*keyword*	specifies style of legend for star vertices
STARLEGENDLAB= *'label'*	specifies label for STARLEGEND= legend
STAROUTRADIUS=*value*	specifies outer radius of stars
STARSPEC=*value*| *SAS-data-set*	specifies method used to standardize vertex variables
STARSTART=*value*	specifies angle for first vertex
STARTYPE=*keyword*	specifies graphical style of star
STARVERTICES= *variable*|*(variables)*	superimposes star at each point on chart
WSTARCIRCLES=*n*	specifies width of circles requested by the STARCIRCLES= option
WSTARS=*n*	specifies width of stars requested by the STARVERTICES= option

Table 44.21. Overlay Options

CCOVERLAY=*color-list*	specifies colors for overlay line segments
COVERLAY=*color-list*	specifies colors for overlay plots
COVERLAYCLIP=*color*	specifies color for clipped points on overlays
LOVERLAY=*linetypes*	specifies line types for overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=*variable-list*	specifies variables to overlay on control chart
OVERLAYCLIPSYM=*symbol*	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=*value*	specifies symbol height for clipped points on overlays
OVERLAYHTML=*variable-list*	specifies URLS to associate with overlay points
OVERLAYID=*variable-list*	specifies labels for overlay points
OVERLAYLEGLAB='*label*'	specifies label for overlay legend
OVERLAYSYM=*symbol-list*	specifies symbols for overlay plots
OVERLAYSYMHT=*value-list*	specifies symbol heights for overlay plots
WOVERLAY=*value-list*	specifies widths of overlay line segments

Details

Constructing Charts for Number Nonconforming (np Charts)

The following notation is used in this section:

p	expected proportion of nonconforming items produced by the process
p_i	proportion of nonconforming items in the i^{th} subgroup
X_i	number of nonconforming items in the i^{th} subgroup
n_i	number of items in the i^{th} subgroup
$\bar{p}$	average proportion of nonconforming items taken across subgroups: $$\bar{p} = \frac{n_1 p_1 + \cdots + n_N p_N}{n_1 + \cdots + n_N} = \frac{X_1 + \cdots + X_N}{n_1 + \cdots + n_N}$$
N	number of subgroups
$I_T(\alpha, \beta)$	incomplete beta function: $$I_T(\alpha,\beta) = (\Gamma(\alpha+\beta)/\Gamma(\alpha)\Gamma(\beta)) \int_0^T t^{\alpha-1}(1-t)^{\beta-1}dt$$ for $0 < T < 1$, $\alpha > 0$, and $\beta > 0$, where $\Gamma(\cdot)$ is the gamma function

Plotted Points

Each point on an np chart represents the observed number (X_i) of nonconforming items in a subgroup. For example, suppose the first subgroup (see Figure 44.9) contains 12 items, of which three are nonconforming. The point plotted for the first subgroup is $X_1 = 3$.

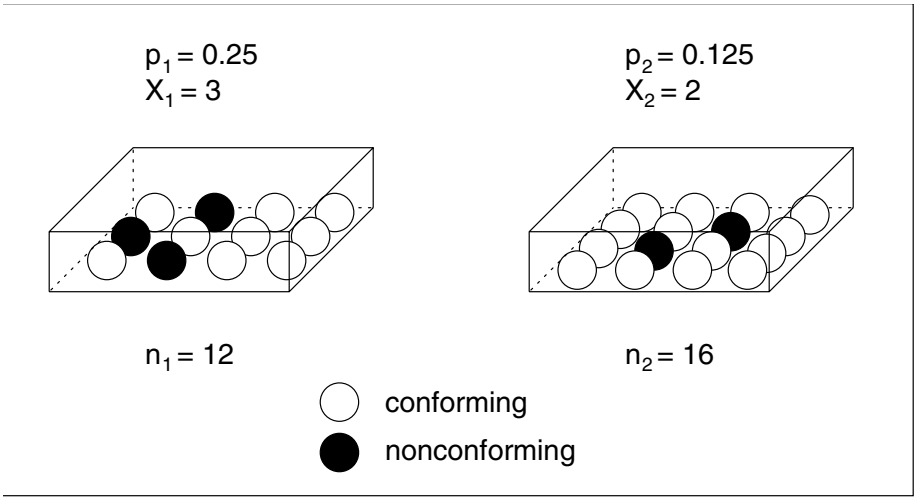


Figure 44.9. Proportions Versus Counts

Note that a p chart displays the proportion of nonconforming items p_i. You can use the PCHART statement to create p charts; see Chapter 45, "PCHART Statement."

Central Line

By default, the central line on an np chart indicates an estimate for $n_i p$, which is computed as $n_i \bar{p}$. If you specify a known value (p_0) for p, the central line indicates the value of $n_i p_0$. Note that the central line varies with n_i.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of X_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that X_i exceeds the limits

The lower and upper control limits, LCL and UCL respectively, are computed as

$$\text{LCL} = \max\left(n_i\bar{p} - k\sqrt{n_i\bar{p}(1-\bar{p})}\ , 0\right)$$
$$\text{UCL} = \min\left(n_i\bar{p} + k\sqrt{n_i\bar{p}(1-\bar{p})}\ , n_i\right)$$

A lower probability limit for X_i can be determined using the fact that

$$\begin{aligned} P\{X_i < \text{LCL}\} &= 1 - P\{X_i \geq \text{LCL}\} \\ &= 1 - I_{\bar{p}}(\text{LCL}, n_i + 1 - \text{LCL}) \\ &= I_{1-\bar{p}}(n_i + 1 - \text{LCL}, \text{LCL}) \end{aligned}$$

Refer to Johnson, Kotz, and Kemp (1992). This assumes that the process is in statistical control and that X_i is binomially distributed. The lower probability limit LCL is then calculated by setting

$$I_{1-\bar{p}}(n_i + 1 - \text{LCL}, \text{LCL}) = \alpha/2$$

and solving for LCL. Similarly, the upper probability limit for X_i can be determined using the fact that

$$\begin{aligned} P\{X_i > \text{UCL}\} &= P\{X_i > \text{UCL}\} \\ &= I_{\bar{p}}(\text{UCL}, n_i + 1 - \text{UCL}) \end{aligned}$$

The upper probability limit UCL is then calculated by setting

$$I_{\bar{p}}(\text{UCL}, n_i + 1 - \text{UCL}) = \alpha/2$$

and solving for UCL. The probability limits are asymmetric about the central line. Note that both the control limits and probability limits vary with n_i.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify p_0 with the P0= option or with the variable _P_ in the LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables can be saved:

Table 44.22. OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLNP	lower control limit for number of nonconforming items
LIMITN	sample size associated with the control limits
_NP_	average number of nonconforming items ($n_i\bar{p}$ or $n_i p_0$)
_P_	average proportion of nonconforming items ($\bar{p}$ or p_0)
SIGMAS	multiple (k) of standard error of X_i
SUBGRP	*subgroup-variable* specified in the NPCHART statement
TYPE	type (standard or estimate) of _NP_
UCLNP	upper control limit for number of nonconforming items
VAR	*process* specified in the NPCHART statement

Notes:

1. If the control limits vary with subgroup sample size, the special missing value *V* is assigned to the variables _LIMITN_, _LCLNP_, _UCLNP_, _NP_, and _SIGMAS_.
2. If the limits are defined in terms of a multiple k of the standard error of X_i, the value of _ALPHA_ is computed as $\alpha = P\{X_i < \text{\_LCLNP\_}\} + P\{X_i > \text{\_UCLNP\_}\}$, using the incomplete beta function.
3. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = (\text{\_UCLNP\_} - \text{\_NP\_})/\sqrt{\text{\_NP\_}(1 - \text{\_NP\_})/\text{\_LIMITN\_}}$. If _LIMITN_ has the special missing value *V*, this value is assigned to _SIGMAS_.
4. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the NPCHART statement. For an example, see "Saving Control Limits" on page 1489.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables are saved:

- the *subgroup-variable*
- the subgroup proportion of nonconforming items variable named by the *process* suffixed with *P*
- a subgroup sample size variable named by the *process* suffixed with *N*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the NPCHART statement. For example, consider the following statements:

```
proc shewhart data=input;
   npchart (rework rejected)*batch / outhistory=summary
                                     subgroupn =30;
run;
```

The data set SUMMARY contains variables named BATCH, REWORKP, REWORKN, REJETEDP, and REJETEDN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Proportions of Nonconforming Items" on page 1488.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables are saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on np chart
LCLNP	lower control limit for number of nonconforming items
LIMITN	nominal sample size associated with the control limits
SIGMAS	multiple (k) of the standard error of X_i associated with the control limits
subgroup	values of the subgroup variable
SUBNP	subgroup number of nonconforming items
SUBN	subgroup sample size
TESTS	tests for special causes signaled on np chart
UCLNP	upper control limit for number of nonconforming items
VAR	*process* specified in the NPCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The k^{th} character of a value of _TESTS_ is k if Test k is positive at that subgroup. For example, if you request the first four tests (the tests appropriate for np charts) and Tests 2 and 4 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, a 4 for the fourth character, and blanks for the other six characters.
3. The variables _EXLIM_ and _TESTS_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1489.

ODS Tables

The following table summarizes the ODS tables that you can request with the NPCHART statement.

Table 44.23. ODS Tables Produced with the NPCHART Statement

Table Name	Description	Options
NPCHART	*np* chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

Input Data Sets

DATA= Data Set

You can read raw data (counts of nonconforming items) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the NPCHART statement must be a SAS variable in the DATA= data set. This variable provides counts for subgroup samples indexed by the values of the *subgroup-variable*. The *subgroup-variable*, which is specified in the NPCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a count for each *process* and a value for the *subgroup-variable*. The data set must contain one observation for each subgroup. Note that you can specify the DATAUNIT= option in the NPCHART statement to read proportions or percentages of nonconforming items instead of counts. Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

When you use a DATA= data set with the NPCHART statement, the SUBGROUPN= option (which specifies the subgroup sample size) is required. By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 1936).

For an example of a DATA= data set, see "Creating np Charts from Count Data" on page 1484.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set CONLIMS:*

```
proc shewhart data=info limits=conlims;
   npchart rejects*batch  / subgroupn=100;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLNP_, _NP_, and _UCLNP_, which specify the control limits directly
- the variable _P_, which is used to calculate the control limits according to the equations on page 1507

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are `ESTIMATE` and `STANDARD`.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1492.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This allows you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART procedure or to create your own HISTORY= data set.

A HISTORY= data set used with the NPCHART statement must contain

- the *subgroup-variable*
- a subgroup proportion of nonconforming items variable for each *process*
- a subgroup sample size variable for each *process*

*In Release 6.09 and in earlier releases, it is necessary to specify the READLIMITS option.

The names of the proportion sample size variables must be the *process* name concatenated with the special suffix characters P and N, respectively.

For example, consider the following statements:

```
proc shewhart history=summary;
   npchart ( rework rejected)*batch / subgroupn=50;
run;
```

The data set SUMMARY must include the variables BATCH, REWORKP, REWORKN, REJETEDP, and REJETEDN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 1936 for an example).

For an example of a HISTORY= data set, see "Creating np Charts from Summary Data" on page 1486.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information read from a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in Chapter 56, "Specialized Control Charts."

The following table lists the variables required in a TABLE= data set used with the NPCHART statement:

Table 44.24. Variables Required in a TABLE= Data Set

Variable	Description
LCLNP	lower control limit for number of nonconforming items
LIMITN	nominal sample size associated with the control limits
_NP_	average number of nonconforming items
subgroup-variable	values of the *subgroup-variable*
SUBN	subgroup sample size
SUBNP	subgroup number of nonconforming items
UCLNP	upper control limit for number of nonconforming items

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1489.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical	DATA=	*process*
Vertical	HISTORY=	subgroup number nonconforming variable
Vertical	TABLE=	_SUBNP_

For an example, see "Labeling Axes" on page 1966.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples

This section provides advanced examples of the NPCHART statement.

Example 44.1. Applying Tests for Special Causes

See SHWNP2 in the SAS/QC Sample Library

This example shows how you can apply tests for special causes to make *np* charts more sensitive to special causes of variation. The following statements create a SAS data set named CIRCUIT3, which contains the number of failing circuits for 20 batches from the circuit manufacturing process introduced in the "Creating np Charts from Count Data" section on page 1484:

```
data circuit3;
   input batch fail @@;
datalines;
 1 12    2 21    3 16    4  9
 5  3    6  4    7  6    8  9
 9 11   10 13   11 12   12  7
13  2   14 14   15  9   16  8
17 14   18 10   19 11   20  9
;
run;
```

The following statements create the *np* chart, apply several tests to the chart, and tabulate the results:

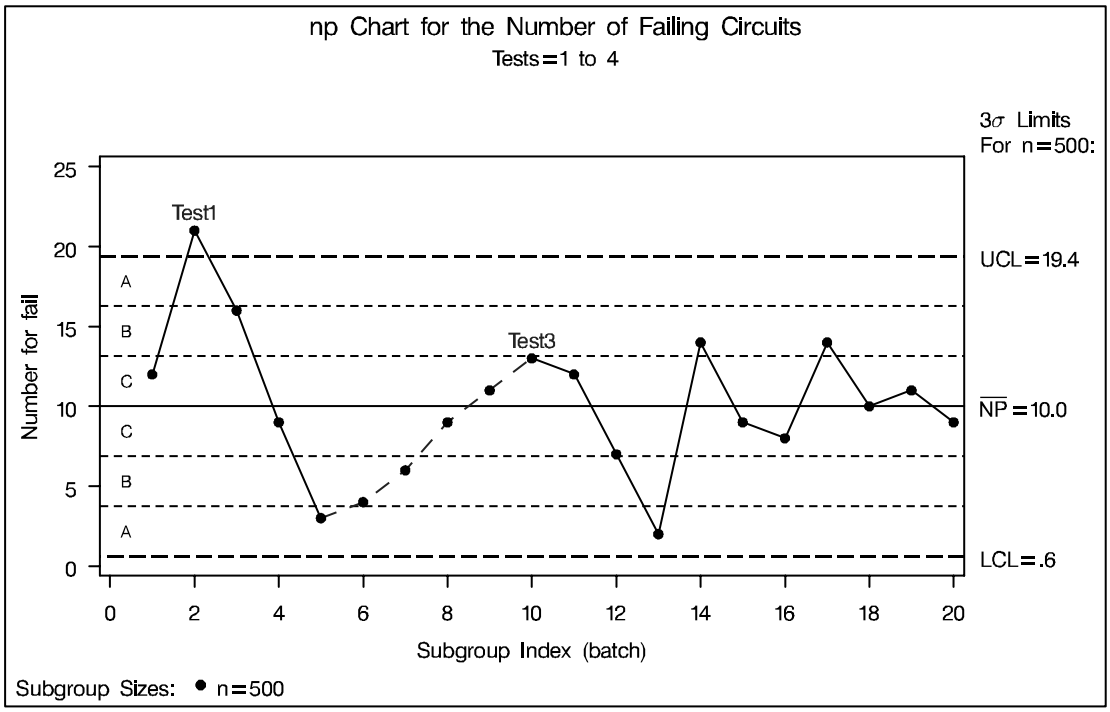
```
symbol h = .8;
title1'np Chart for the Number of Failing Circuits';
title2 'Tests=1 to 4';
proc shewhart data=circuit3;
   npchart fail*batch / subgroupn = 500
                        tests=1 to 4
                        zones
                        zonelabels
                        ltests=20
                        table
                        tabletest
                        tablelegend ;
run;
```

The chart is shown in Output 44.1.1, and the printed output is shown in Output 44.1.2. The TESTS= option requests Tests 1, 2, 3, and 4, which are described in Chapter 55, "Tests for Special Causes." The TABLETESTS option requests a table of counts of nonconforming items and control limits, with a column indicating which subgroups tested positive for special causes. The TABLELEGEND option adds a legend describing the tests.

The ZONELABELS option displays zone lines and zone labels on the chart. The zones are used to define the tests. The LTESTS= option specifies the line type used to connect the points in a pattern for a test that is signaled.

Output 44.1.1 and Output 44.1.2 indicate that Test 1 is positive at batch 2 and Test 3 is positive at batch 10.

Output 44.1.1. Tests for Special Causes Displayed on np Chart



Output 44.1.2. Tabular Form of np Chart

np Chart for the Number of Failing Circuits
Tests=1 to 4

np Chart Summary for fail

batch	Subgroup Sample Size	-3 Sigma Limits with n=500 for Number- Lower Limit	Subgroup Number	Upper Limit	Special Tests Signaled
1	500	0.60851449	12.000000	19.391486	
2	500	0.60851449	21.000000	19.391486	1
3	500	0.60851449	16.000000	19.391486	
4	500	0.60851449	9.000000	19.391486	
5	500	0.60851449	3.000000	19.391486	
6	500	0.60851449	4.000000	19.391486	
7	500	0.60851449	6.000000	19.391486	
8	500	0.60851449	9.000000	19.391486	
9	500	0.60851449	11.000000	19.391486	
10	500	0.60851449	13.000000	19.391486	3
11	500	0.60851449	12.000000	19.391486	
12	500	0.60851449	7.000000	19.391486	
13	500	0.60851449	2.000000	19.391486	
14	500	0.60851449	14.000000	19.391486	
15	500	0.60851449	9.000000	19.391486	
16	500	0.60851449	8.000000	19.391486	
17	500	0.60851449	14.000000	19.391486	
18	500	0.60851449	10.000000	19.391486	
19	500	0.60851449	11.000000	19.391486	
20	500	0.60851449	9.000000	19.391486	

Test Descriptions

Test 1 One point beyond Zone A (outside control limits)
Test 3 Six points in a row steadily increasing or decreasing

Example 44.2. Specifying Standard Average Proportion

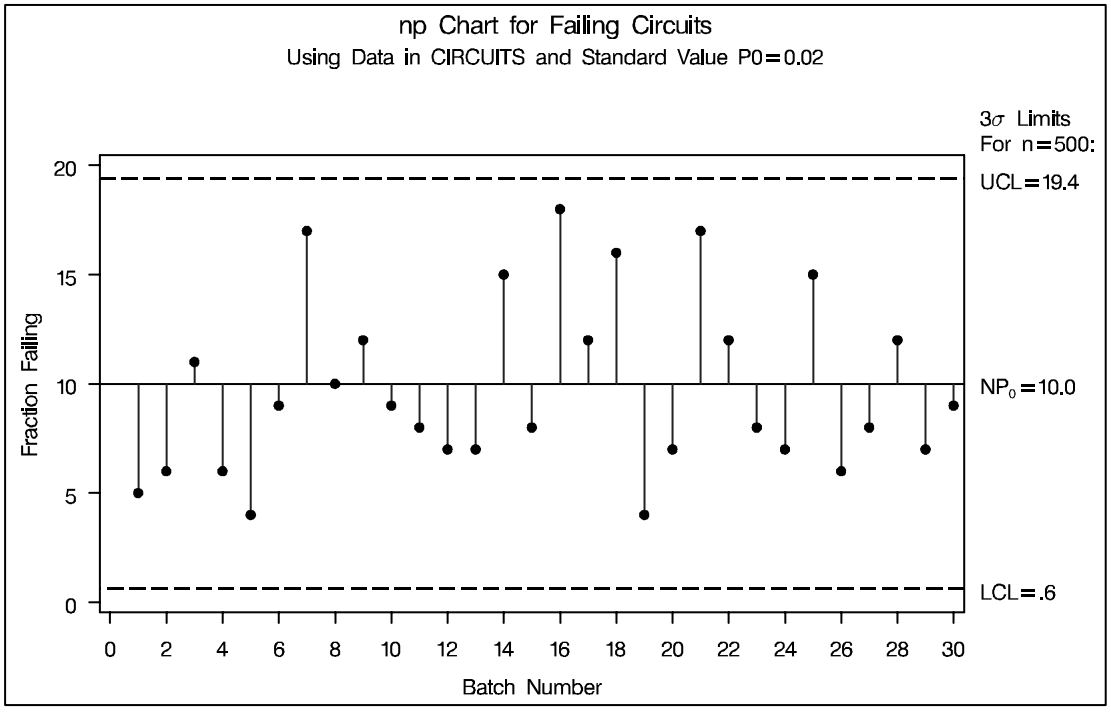
See SHWNP3 in the SAS/QC Sample Library

In some situations, a standard (known) value (p_0) is available for the expected proportion of nonconforming items, based on extensive testing or previous sampling. This example illustrates how you can specify p_0 to create an np chart.

An np chart is used to monitor the number of failing circuits in the data set CIRCUITS, which is introduced on page 1484. The expected proportion of failing circuits is known to be $p_0 = 0.02$. The following statements create an np chart, shown in Output 44.2.1, using p_0 to compute the control limits:

```
symbol h = .8;
title1 'np Chart for Failing Circuits';
title2 'Using Data in CIRCUITS and Standard Value P0=0.02';
proc shewhart data=circuits;
   npchart fail*batch / subgroupn = 500
                        p0         = 0.02
                        npsymbol   = np0
                        nolegend
                        needles;
   label batch ='Batch Number'
         fail  ='Fraction Failing';
run;
```

Output 44.2.1. An np Chart with Standard Value of p_0



The chart indicates that the process is in control. The P0= option specifies p_0. The NPSYMBOL= option specifies a label for the central line indicating that the line

represents a standard value. The NEEDLES option connects points to the central line with vertical needles. The NOLEGEND option suppresses the default legend for subgroup sample sizes. Labels for the vertical and horizontal axes are provided with the LABEL statement. For details concerning axis labeling, see "Axis Labels" on page 1514.

Alternatively, you can specify p_0 using the variable _P_ in a LIMITS= data set, as follows:

```
data climits;
   length _var_ _subgrp_ _type_ $8;
   _p_      = 0.02;
   _subgrp_ = 'batch';
   _var_    = 'fail';
   _type_   = 'STANDARD';
   _limitn_ = 500;

proc shewhart data=circuits limits=climits;
   npchart fail*batch / subgroupn = 500
                        npsymbol  = np0
                        nolegend
                        needles;
   label batch ='Batch Number'
         fail  ='Fraction Failing';
run;
```

The bookkeeping variable _TYPE_ indicates that _P_ has a standard value. The chart produced by these statements is identical to the chart in Output 44.2.1.

Example 44.3. Working with Unequal Subgroup Sample Sizes

See SHWNP4 in the SAS/QC Sample Library

The following statements create a SAS data set named BATTERY, which contains the number of alkaline batteries per lot failing an acceptance test. The number of batteries tested in each lot varies but is approximately 150.

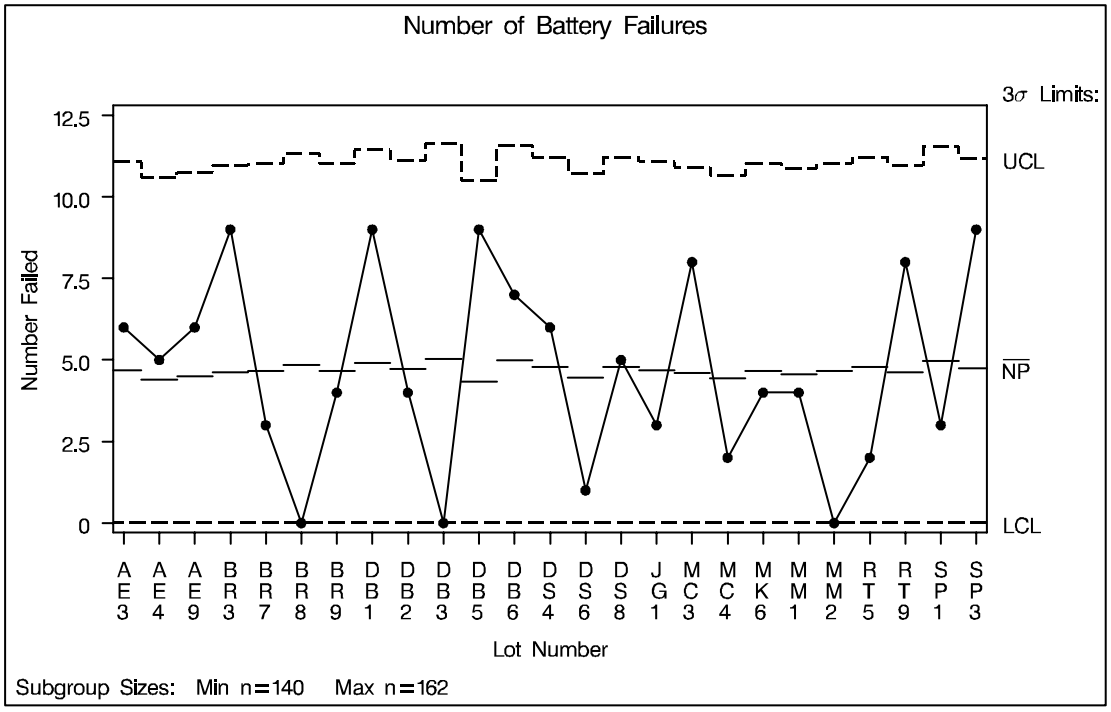
```
data battery;
   length lot $3;
   input lot nfailed sampsize @@;
   label nfailed ='Number Failed'
         lot     = 'Lot Number'
         sampsize='Number Sampled';
   datalines;
AE3  6  151     AE4  5  142     AE9  6  145
BR3  9  149     BR7  3  150     BR8  0  156
BR9  4  150     DB1  9  158     DB2  4  152
DB3  0  162     DB5  9  140     DB6  7  161
DS4  6  154     DS6  1  144     DS8  5  154
JG1  3  151     MC3  8  148     MC4  2  143
MK6  4  150     MM1  4  147     MM2  0  150
RT5  2  154     RT9  8  149     SP1  3  160
SP3  9  153
;
run;
```

The variable NFAILED contains the number of battery failures, the variable LOT contains the lot number, and the variable SAMPSIZE contains the lot sample size. The following statements request an np chart for this data:

```
title 'Number of Battery Failures';
proc shewhart data=battery;
   npchart nfailed*lot / subgroupn = sampsize
                         outlimits = batlim
                         turnhlabels;
   label nfailed='Number Failed';
run;
```

The chart is shown in Output 44.3.1, and the OUTLIMITS= data set BATLIM is listed in Output 44.3.2.

Output 44.3.1. An np Chart with Varying Subgroup Sample Sizes



Note that the upper control limit and central line on the np chart vary with the subgroup sample size. The lower control limit is truncated at zero. The sample size legend indicates the minimum and maximum subgroup sample sizes.

The variables in BATLIM whose values vary with subgroup sample size are assigned the special missing value *V*.

The SHEWHART procedure provides various options for working with unequal subgroup sample sizes. For example, you can use the LIMITN= option to specify a fixed (nominal) sample size for computing the control limits, as illustrated by the following statements:

Output 44.3.2. The Control Limits Data Set BATLIM

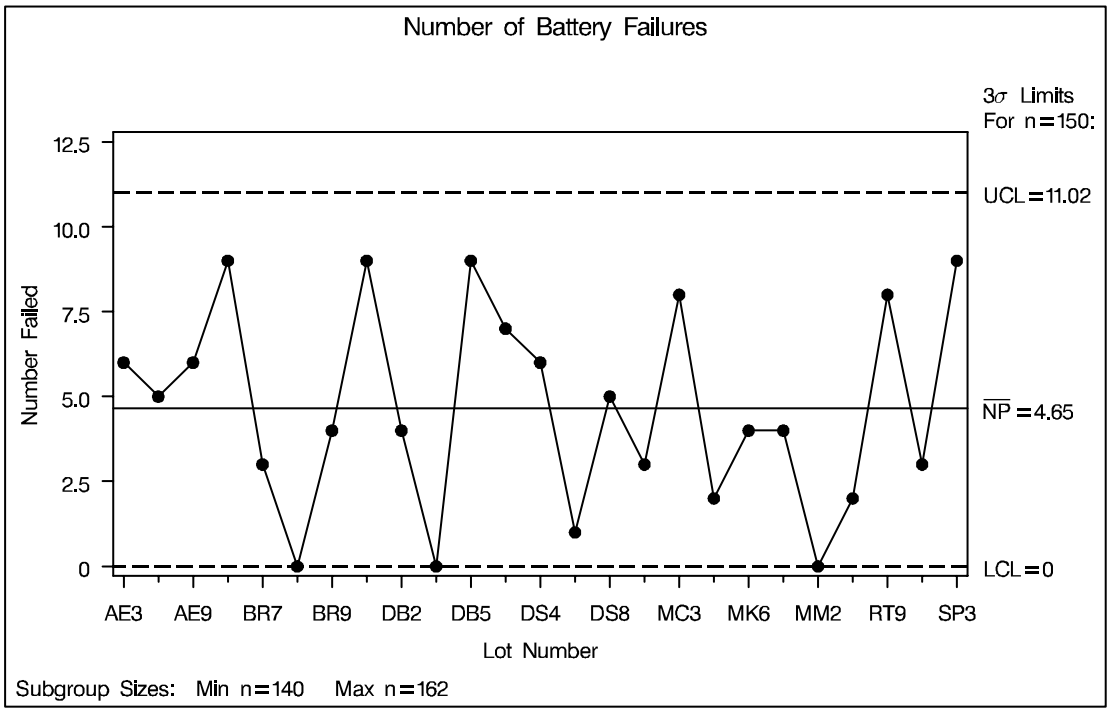
Control Limits for Battery Failures

VAR	_SUBGRP_	_TYPE_	_LIMIT_	_ALPHA_	_SIGMA_	_P_	_LCL_	_NP_	_UCL_
nfailed	lot	ESTIMATE	V	V	3	0.031010	V	V	V

```
title 'Number of Battery Failures';
proc shewhart data=battery;
   npchart nfailed*lot / subgroupn = sampsize
                         limitn    = 150
                         alln;
   label nfailed='Number Failed';
run;
```

The ALLN option specifies that all points (regardless of subgroup sample size) are to be displayed. By default, only points for subgroups whose sample size matches the LIMITN= value are displayed. The chart is shown in Output 44.3.3.

Output 44.3.3. Control Limits Based on Fixed Subgroup Sample Size



All the points are inside the control limits, indicating that the process is in statistical control. Since there is relatively little variation in the sample sizes, the control limits in Output 44.3.3 provide a close approximation to the exact control limits in Output 44.3.1, and the same conclusions can be drawn from both charts. In general, you should be careful when interpreting charts that use a nominal sample size to compute control limits, since these limits are only approximate when the sample sizes vary.

Example 44.4. Specifying Control Limit Information

See SHWNP5 in the SAS/QC Sample Library

This example shows how to use the DATA step to create LIMITS= data sets for use with the NPCHART statement. The variables _VAR_ and _SUBGRP_ are required. These variables must be character variables whose lengths are no greater than 32, and their values must match the *process* and *subgroup-variable* specified in the NPCHART statement. In addition, you must provide one of the following:

- the variables _LCLNP_, _NP_, and _UCLNP_
- the variable _P_

The following DATA step creates a data set named CLIMITS1, which provides a complete set of control limits for an np chart:

```
data climits1;
   length _var_ _subgrp_ _type_ $8;
   _var_    = 'fail';
   _subgrp_ = 'batch';
   _limitn_ = 500;
   _type_   = 'STANDARD';
   _lclnp_  =  0;
   _np_     = 10;
   _uclnp_  = 20;
run;
```

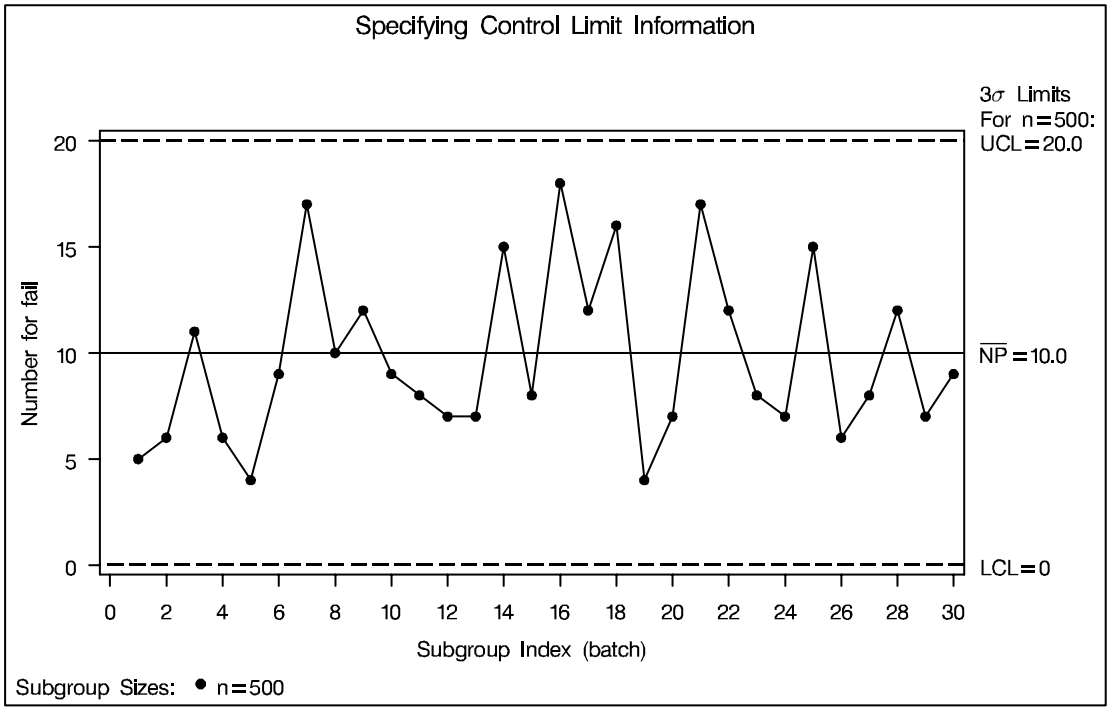
The following statements read the control limits* from the data set CLIMITS1 and apply them to the count data in the data set CIRCUITS, which is introduced on page 1484:

```
title 'Specifying Control Limit Information';
proc shewhart data=circuits limits=climits1;
   npchart fail*batch / subgroupn = 500;
run;
```

The chart is shown in Output 44.4.1.

*In Release 6.09 and in earlier releases, you must also specify the READLIMITS option.

Output 44.4.1. Control Limit Information Read from CLIMITS1

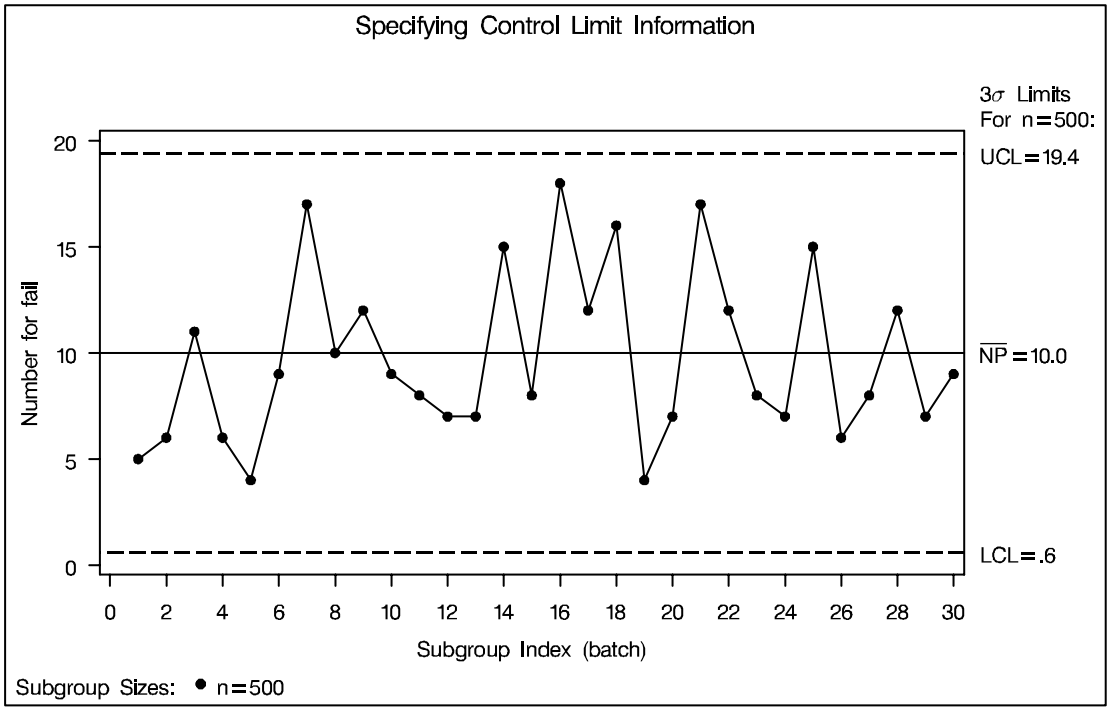


The following DATA step creates a data set named CLIMITS2, which provides a value for the expected proportion of nonconforming items (_P_). This parameter is then used to compute the control limits for the data in CIRCUITS according to the equations on page 1507.

```
data climits2;
   length _var_ _subgrp_ _type_ $8;
   _var_    = 'fail';
   _subgrp_ = 'batch';
   _limitn_ = 500;
   _type_   = 'STANDARD';
   _p_      = .02;
run;


title 'Specifying Control Limit Information';
proc shewhart data=circuits limits=climits2;
   npchart fail*batch / subgroupn = 500;
run;
```

The chart is shown in Output 44.4.2. Note that the control limits are not the same as those shown in Output 44.4.1.

Output 44.4.2. Control Limit Information Read from CLIMITS2



Chapter 45
PCHART Statement

Chapter Contents

OVERVIEW 1527

GETTING STARTED 1528
- Creating p Charts from Count Data 1528
- Creating p Charts from Summary Data 1530
- Saving Proportions of Nonconforming Items 1533
- Saving Control Limits 1534
- Reading Preestablished Control Limits 1536

SYNTAX 1537
- Summary of Options 1539

DETAILS 1549
- Constructing Charts for Proportion Nonconforming (p Charts) 1549
- Output Data Sets 1551
- ODS Tables 1554
- Input Data Sets 1554
- Axis Labels 1557
- Missing Values 1558

EXAMPLES 1558
- Example 45.1. Applying Tests for Special Causes 1558
- Example 45.2. Specifying Standard Average Proportion 1560
- Example 45.3. Working with Unequal Subgroup Sample Sizes 1562
- Example 45.4. Creating a Chart with Revised Control Limits 1565
- Example 45.5. OC Curve for Chart 1567

Chapter 45
PCHART Statement

Overview

The PCHART statement creates p charts for the proportions of nonconforming (defective) items in subgroup samples.

You can use options in the PCHART statement to

- compute control limits from the data based on a multiple of the standard error of the proportions or as probability limits
- tabulate subgroup sample sizes, proportions of nonconforming items, control limits, and other information
- save control limits in an output data set
- save subgroup sample sizes and proportions of nonconforming items in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify a known (standard) proportion of nonconforming items for computing control limits
- specify the data as counts, proportions, or percentages of nonconforming items
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

Getting Started

This section introduces the PCHART statement with simple examples that illustrate commonly used options. Complete syntax for the PCHART statement is presented in the "Syntax" section on page 1537, and advanced examples are given in the "Examples" section on page 1558.

Creating p Charts from Count Data

See SHWPCHR in the SAS/QC Sample Library

An electronics company manufactures circuits in batches of 500 and uses a p chart to monitor the proportion of failing circuits. Thirty batches are examined, and the failures in each batch are counted. The following statements create a SAS data set named CIRCUITS,* which contains the failure counts:

```
data circuits;
  input batch fail @@;
  datalines;
1    5   2   6   3  11   4   6    5   4
6    9   7  17   8  10   9  12   10   9
11   8  12   7  13   7  14  15   15   8
16  18  17  12  18  16  19   4   20   7
21  17  22  12  23   8  24   7   25  15
26   6  27   8  28  12  29   7   30   9
;
run;
```

A partial listing of CIRCUITS is shown in Figure 45.1.

Number of Failing Circuits

batch	fail
1	5
2	6
3	11
4	6
5	4
.	.
.	.
.	.

Figure 45.1. The Data Set CIRCUITS

There is a single observation for each batch. The variable BATCH identifies the subgroup sample and is referred to as the *subgroup-variable*. The variable FAIL contains the number of nonconforming items in each subgroup sample and is referred to as the *process variable* (or *process* for short).

The following statements create the p chart shown in Figure 45.2:

*This data set is also used in the "Getting Started" section of Chapter 44, "NPCHART Statement."

```
title 'p Chart for the Proportion of Failing Circuits';
proc shewhart data=circuits;
   pchart fail*batch / subgroupn = 500;
run;
```

This example illustrates the basic form of the PCHART statement. After the keyword PCHART, you specify the *process* to analyze (in this case, FAIL), followed by an asterisk and the *subgroup-variable* (BATCH).

The input data set is specified with the DATA= option in the PROC SHEWHART statement. The SUBGROUPN= option specifies the number of items in each subgroup sample and is required with a DATA= input data set. The SUBGROUPN= option specifies one of the following:

- a constant subgroup sample size (as in this case)
- a variable in the input data set whose values provide the subgroup sample sizes (see the next example)

Options such as SUBGROUPN= are specified after the slash (/) in the PCHART statement. A complete list of options is presented in the "Syntax" section on page 1537.

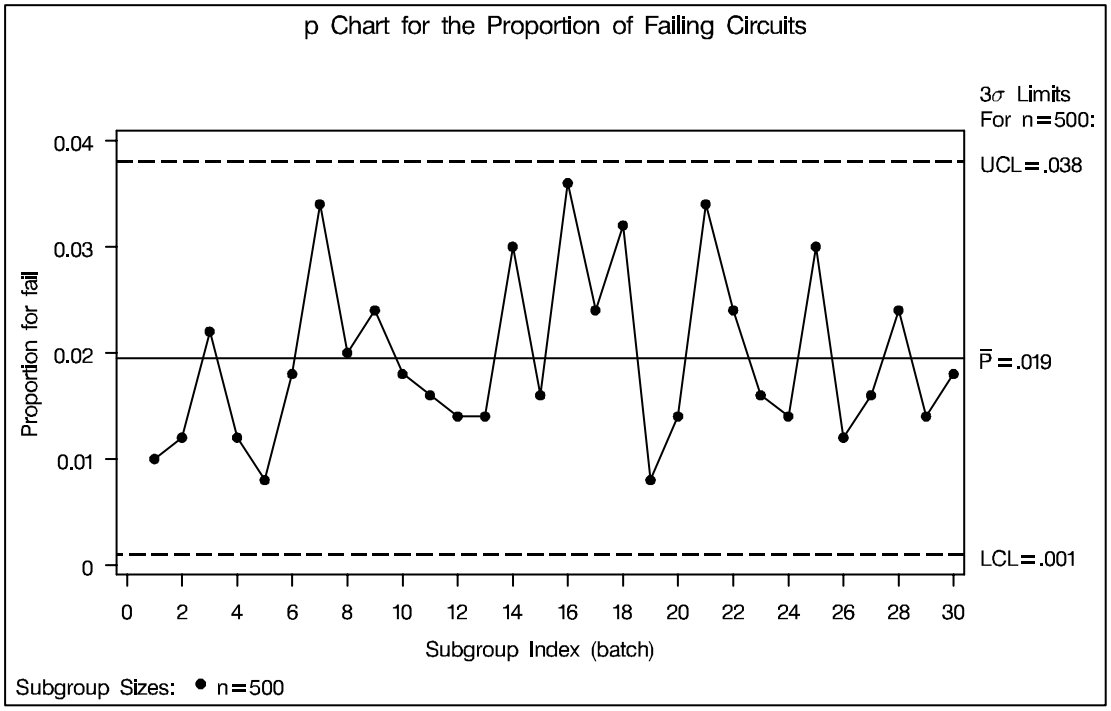


Figure 45.2. A p Chart for Circuit Failures

Each point on the p chart represents the proportion of nonconforming items for a particular subgroup. For instance, the value plotted for the first batch is $5/500 = 0.01$, as illustrated in Figure 45.3.

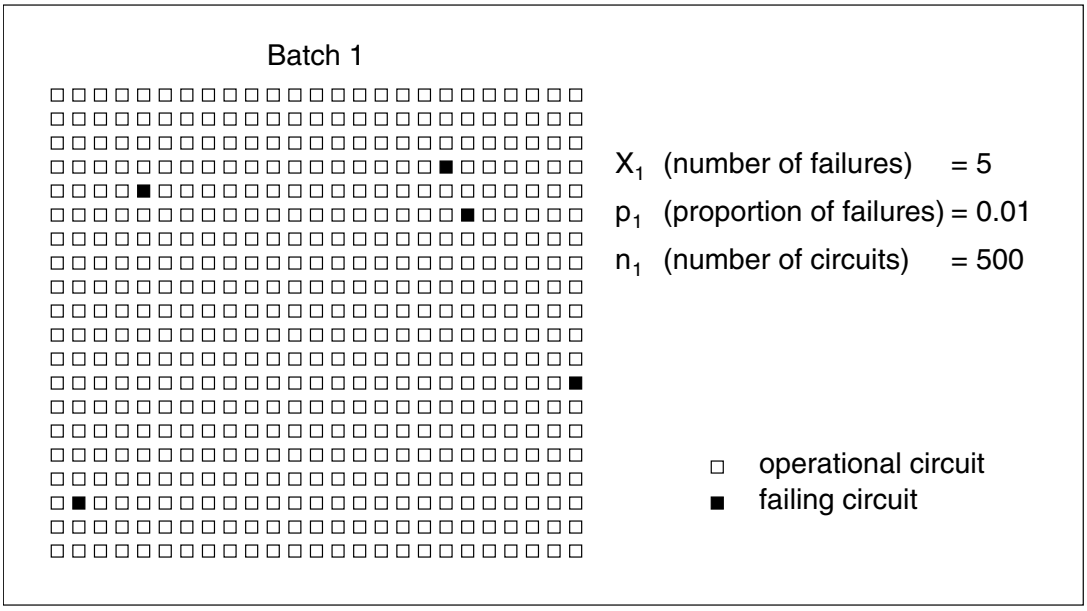


Figure 45.3. Proportions Versus Counts

Since all the points fall within the control limits, it can be concluded that the process is in statistical control.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in "Control Limits" on page 1550. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1536. For computational details, see "Constructing Charts for Proportion Nonconforming (p Charts)" on page 1549. For more details on reading counts of nonconforming items, see "DATA= Data Set" on page 1554.

Creating p Charts from Summary Data

See SHWPCHR in the SAS/QC Sample Library

The previous example illustrates how you can create p charts using raw data (counts of nonconforming items). However, in many applications, the data are provided in summarized form as proportions or percentages of nonconforming items. This example illustrates how you can use the PCHART statement with data of this type.

The following data set provides the data from the preceding example in summarized form:

```
data cirprop;
   input batch pfailed @@;
   sampsize=500;
datalines;
 1  0.010   2  0.012   3  0.022   4  0.012   5  0.008
 6  0.018   7  0.034   8  0.020   9  0.024  10  0.018
11  0.016  12  0.014  13  0.014  14  0.030  15  0.016
16  0.036  17  0.024  18  0.032  19  0.008  20  0.014
21  0.034  22  0.024  23  0.016  24  0.014  25  0.030
26  0.012  27  0.016  28  0.024  29  0.014  30  0.018
;
run;
```

A partial listing of CIRPROP is shown in Figure 45.4. The subgroups are still indexed by BATCH. The variable PFAILED contains the proportions of nonconforming items, and the variable SAMPSIZE contains the subgroup sample sizes.

Subgroup Proportions of Nonconforming Items

batch	pfailed	sampsize
1	0.010	500
2	0.012	500
3	0.022	500
4	0.012	500
5	0.008	500
.	.	.
.	.	.
.	.	.

Figure 45.4. The Data Set CIRPROP

The following statements create a p chart identical to the one in Figure 45.2:

```
title 'p Chart for the Proportion of Failing Circuits';
symbol v=dot;
proc shewhart data=cirprop;
   pchart pfailed*batch / subgroupn=sampsize
                          dataunit =proportion;
label pfailed = 'Proportion for FAIL';
run;
```

The DATAUNIT= option specifies that the values of the *process* (PFAILED) are proportions of nonconforming items. By default, the values of the *process* are assumed to be counts of nonconforming items (see the previous example).

Alternatively, you can read the data set CIRPROP by specifying it as a HISTORY= data set in the PROC SHEWHART statement. A HISTORY= data set used with the PCHART statement must contain the following variables:

- subgroup variable
- subgroup proportion of nonconforming items variable
- subgroup sample size variable

Furthermore, the names of the subgroup proportion and sample size variables must begin with the *process* name specified in the PCHART statement and end with the special suffix characters P and N, respectively.

To specify CIRPROP as a HISTORY= data set and FAIL as the *process*, you must rename the variables PFAILED and SAMPSIZE to FAILP and FAILN, respectively. The following statements temporarily rename PFAILED and SAMPSIZE for the duration of the procedure step:

```
options ls=84 ps=30;
title 'p Chart for the Proportion of Failing Circuits';
proc shewhart history=cirprop(rename=(pfailed=failp
                                sampsize=failn )) lineprinter;
   pchart fail*batch='*';
run;
options ls=76 ps=80;
```

The resulting p chart is shown in Figure 45.5. Since the LINEPRINTER option is specified in the PROC SHEWHART statement, line printer output is produced. * The asterisk specified in single quotes after the *subgroup-variable* indicates the character used to plot points. This character must follow an equal sign.

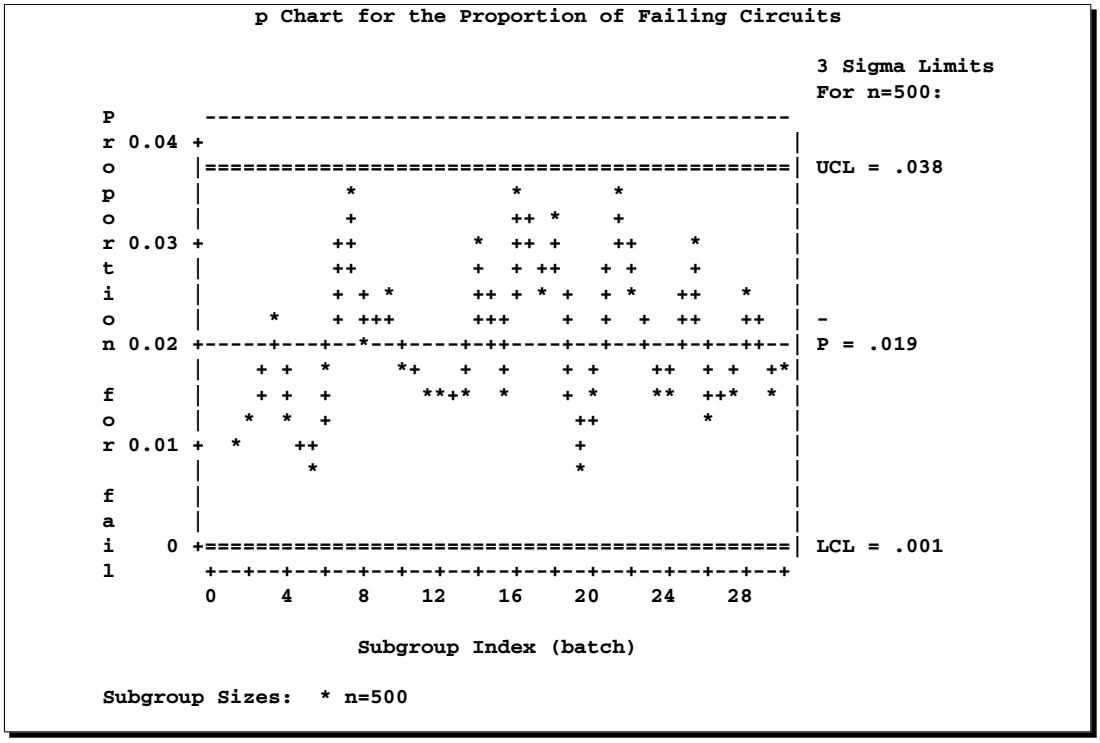


Figure 45.5. A p Chart from Subgroup Proportions

In this example, it is more convenient to use CIRPROP as a DATA= data set than as a HISTORY= data set. In general, it is more convenient to use the HISTORY= option for input data sets that have been previously created by the SHEWHART procedure as OUTHISTORY= data sets, as illustrated in the next example. For more information, see "HISTORY= Data Set" on page 1555.

*In Release 6.12 and previous releases of SAS/QC software, the keyword GRAPHICS was required in the PROC SHEWHART statement to specify that the chart be created with a graphics device. In Version 7, you can specify the LINEPRINTER option to request line printer plots.

Saving Proportions of Nonconforming Items

See SHWPCHR in the SAS/QC Sample Library

In this example, the PCHART statement is used to create a data set that can later be read by the SHEWHART procedure (as in the preceding example). The following statements read the number of nonconforming items from the data set CIRCUITS (see page 1528) and create a summary data set named CIRHIST:

```
proc shewhart data=circuits;
   pchart fail*batch / subgroupn  = 500
                       outhistory = cirhist
                       nochart ;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 45.2. Figure 45.6 contains a partial listing of CIRHIST.

Subgroup Proportions and Control Limit Information

batch	failP	fail N
1	0.010	500
2	0.012	500
3	0.022	500
4	0.012	500
.	.	.
.	.	.
.	.	.

Figure 45.6. The Data Set CIRHIST

There are three variables in the data set CIRHIST.

- BATCH contains the subgroup index.
- FAILP contains the subgroup proportion of nonconforming items.
- FAILN contains the subgroup sample size.

Note that the variables containing the subgroup proportions of nonconforming items and subgroup sample sizes are named by adding the suffix characters *P* and *N* to the *process* FAIL specified in the PCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets. For more information, see "OUTHISTORY= Data Set" on page 1552.

Saving Control Limits

See SHWPCHR in the SAS/QC Sample Library

You can save the control limits for a p chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1536) or modify the limits with a DATA step program.

The following statements read the number of nonconforming items per subgroup from the data set CIRCUITS (see page 1528) and save the control limits displayed in Figure 45.2 in a data set named CIRLIM:

```
proc shewhart data=circuits;
   pchart fail*batch / subgroupn = 500
                       outlimits = cirlim
                       nochart ;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. The data set CIRLIM is listed in Figure 45.7.

Control Limits for the Proportion of Failing Circuits

VARIABLE	_SUBGRP_	_TYPE_	_LIMITN_	_LCLP_	_SIGMAS_	_P_	_UCLP_
fail	batch	ESTIMATE	500	.005040334	3	.000930786	0.019467	0.038003

Figure 45.7. The Data Set CIRLIM Containing Control Limit Information

The data set CIRLIM contains one observation with the limits for *process* FAIL. The variables _LCLP_ and _UCLP_ contain the lower and upper control limits, and the variable _P_ contains the central line. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the value of _P_ is an estimate or standard value.

For more information, see "OUTLIMITS= Data Set" on page 1551.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=circuits;
   pchart fail*batch / subgroupn = 500
                       outtable  = cirtable
                       nochart ;
run;
```

The data set CIRTABLE is listed in Figure 45.8.

Subgroup Proportions and Control Limit Information

VAR	batch	_SIGMAS_	_LIMITN_	_SUBN_	_LCLP_	_SUBP_	_P_	_UCLP_	_EXLIM_
fail	1	3	500	500	.000930786	0.010	0.019467	0.038003	
fail	2	3	500	500	.000930786	0.012	0.019467	0.038003	
fail	3	3	500	500	.000930786	0.022	0.019467	0.038003	
fail	4	3	500	500	.000930786	0.012	0.019467	0.038003	
fail	5	3	500	500	.000930786	0.008	0.019467	0.038003	
fail	6	3	500	500	.000930786	0.018	0.019467	0.038003	
fail	7	3	500	500	.000930786	0.034	0.019467	0.038003	
fail	8	3	500	500	.000930786	0.020	0.019467	0.038003	
fail	9	3	500	500	.000930786	0.024	0.019467	0.038003	
fail	10	3	500	500	.000930786	0.018	0.019467	0.038003	
fail	11	3	500	500	.000930786	0.016	0.019467	0.038003	
fail	12	3	500	500	.000930786	0.014	0.019467	0.038003	
fail	13	3	500	500	.000930786	0.014	0.019467	0.038003	
fail	14	3	500	500	.000930786	0.030	0.019467	0.038003	
fail	15	3	500	500	.000930786	0.016	0.019467	0.038003	
fail	16	3	500	500	.000930786	0.036	0.019467	0.038003	
fail	17	3	500	500	.000930786	0.024	0.019467	0.038003	
fail	18	3	500	500	.000930786	0.032	0.019467	0.038003	
fail	19	3	500	500	.000930786	0.008	0.019467	0.038003	
fail	20	3	500	500	.000930786	0.014	0.019467	0.038003	
fail	21	3	500	500	.000930786	0.034	0.019467	0.038003	
fail	22	3	500	500	.000930786	0.024	0.019467	0.038003	
fail	23	3	500	500	.000930786	0.016	0.019467	0.038003	
fail	24	3	500	500	.000930786	0.014	0.019467	0.038003	
fail	25	3	500	500	.000930786	0.030	0.019467	0.038003	
fail	26	3	500	500	.000930786	0.012	0.019467	0.038003	
fail	27	3	500	500	.000930786	0.016	0.019467	0.038003	
fail	28	3	500	500	.000930786	0.024	0.019467	0.038003	
fail	29	3	500	500	.000930786	0.014	0.019467	0.038003	
fail	30	3	500	500	.000930786	0.018	0.019467	0.038003	

Figure 45.8. The Data Set CIRTABLE

This data set contains one observation for each subgroup sample. The variables _SUBP_ and _SUBN_ contain the subgroup proportions of nonconforming items and subgroup sample sizes. The variables _LCLP_ and _UCLP_ contain the lower and upper control limits, and the variable _P_ contains the central line. The variables _VAR_ and BATCH contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1553.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read the information in CIRTABLE and display a p chart (not shown here) identical to the chart in Figure 45.2:

```
title 'p Chart for the Proportion of Failing Circuits';
proc shewhart table=cirtable;
   pchart fail*batch;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see Chapter 56, "Specialized Control Charts,"). For more information, see "TABLE= Data Set" on page 1556.

Reading Preestablished Control Limits

See SHWPCHR in the SAS/QC Sample Library

In the previous example, the OUTLIMITS= data set CIRLIM saved control limits computed from the data in CIRCUITS. This example shows how these limits can be applied to new data provided in the following data set:

```
data circuit2;
   input batch fail @@;
datalines;
31  12   32   9   33  16   34   9
35   3   36   8   37  20   38   4
39   8   40   6   41  12   42  16
43   9   44   2   45  10   46   8
47  14   48  10   49  11   50   9
;
run;
```

The following statements create a p chart for the data in CIRCUIT2 using the control limits in CIRLIM:

```
title 'p Chart for the Proportion of Failing Circuits';
proc shewhart data=circuit2 limits=cirlim;
   pchart fail*batch / subgroupn = 500;
run;
```

The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default,* this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name FAIL
- the value of _SUBGRP_ matches the *subgroup-variable* name BATCH

The resulting p chart is shown in Figure 45.9.

The proportion of nonconforming items in the 37^{th} batch exceeds the upper control limit, signaling that the process is out of control.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1555 for details concerning the variables that you must provide.

*In Release 6.09 and in earlier releases, it is also necessary to specify the READLIMITS option to read control limits from a LIMITS= data set.

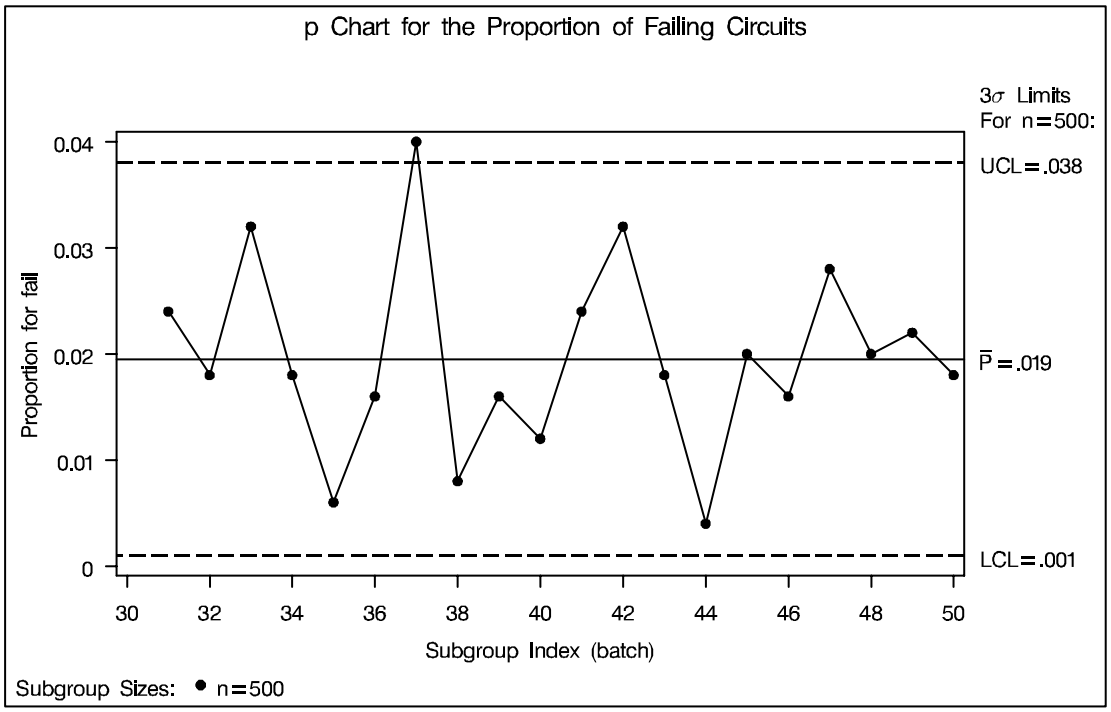


Figure 45.9. A p Chart for Second Set of Circuit Failures

Syntax

The basic syntax for the PCHART statement is as follows:

PCHART *process***subgroup-variable* **;**

The general form of this syntax is as follows:

PCHART (*processes***)****subgroup-variable* <**(***block-variables* **)** >
< =*symbol-variable* | =’*character*’ > < **/** *options* >**;**

You can use any number of PCHART statements in the SHEWHART procedure. The components of the PCHART statement are described as follows.

process
processes
identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If numbers of nonconforming items are read from a DATA= data set, *process* must be the name of the variable containing the numbers. For an example, see “Creating p Charts from Summary Data” on page 1530.
- If proportions of nonconforming items are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see “Creating p Charts from Summary Data” on page 1530.

- If proportions of nonconforming items and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1534.

A *process* is required. If you specify more than one process, enclose the list in parentheses. For example, the following statements request distinct p charts for REJECTS and REWORKS:

```
proc shewhart data=measures;
   pchart (rejects reworks)*sample / subgroupn=100;
run;
```

Note that when data are read from a DATA= data set, the SUBGROUPN= option, which specifies subgroup sample sizes, is required.

subgroup-variable
is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding PCHART statement, SAMPLE is the subgroup variable. For details, see "Subgroup Variables" on page 1771.

block-variables
are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 1932 for an example.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker or character used to plot proportions of nonconforming items.

- If you produce a chart on a line printer, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce a chart on a graphics device, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOLn statements. See "Displaying Stratification in Levels of a Classification Variable" on page 1931 for an example.

character
specifies a plotting character for charts produced on line printers. For example, the following statements create a p chart using an asterisk (*) to plot the points:

```
proc shewhart data=values;
   pchart rejects*day='*' / subgroupn=100;
run;
```

options
enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The "Summary of Options" section, which follows, lists all options by function. Chapter 53, "Dictionary of Options," describes each option in detail.

Summary of Options

The following tables list the PCHART statement options by function. For complete descriptions, see Chapter 53, "Dictionary of Options."

Table 45.1. Tabulation Options

TABLE	creates a basic table of subgroup sample sizes, subgroup proportions of nonconforming items, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with a column indicating which tests for special causes are positive

Note that specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.

Table 45.2. Reference Line Options

CHREF=*color*	specifies color for lines requested by the HREF= option
CVREF=*color*	specifies color for lines requested by the VREF= option
HREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis
HREFCHAR='*character*'	specifies line character for HREF= lines
HREFDATA= *SAS-data-set*	specifies position of reference lines perpendicular to horizontal
HREFLABELS= *'label1'...'labeln'*	specifies labels for HREF= lines
HREFLABPOS=*n*	specifies position of HREFLABELS= labels
LHREF=*linetype*	specifies line type for HREF= lines
LVREF=*linetype*	specifies line type for VREF= lines
NOBYREF	specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to vertical axis
VREFCHAR='*character*'	specifies line character for VREF= lines
VREFLABELS= *'label1'...'labeln'*	specifies labels for VREF= lines
VREFLABPOS=*n*	specifies position of VREFLABELS= labels

Table 45.3. Options for Specifying Tests for Special Causes

NO3SIGMACHECK	allows tests to be applied with control limits other than 3σ limits
TESTS=*value-list*| *customized-pattern-list*	specifies tests for special causes
TEST2RUN=*n*	specifies length of pattern for Test 2
TEST3RUN=*n*	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL='*label*'| (*variable*)|*keyword*	provides labels for points where test is positive
TESTLABEL*n*='*label*'	specifies label for n^{th} test for special causes
TESTNMETHOD= STANDARDIZE	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
TESTRESET=*variable*	allows tests for special causes to be reset
ZONELABELS	adds labels A, B, and C to zone lines
ZONES	adds lines delineating zones A, B, and C
ZONEVALPOS=*n*	specifies position of ZONEVALUES labels
ZONEVALUES	labels zone lines with their values

Table 45.4. Graphical Options for Displaying Tests for Special Causes

CTESTLABBOX=*color*	specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=*color*| *test-color-list*	specifies color for labels indicating points where test is positive
CTESTSYMBOL=*color*	specifies color for symbol used to plot points where test is positive
CZONES=*color*	specifies color for lines and labels delineating zones A, B, and C
LTESTS=*linetype*	specifies type of line connecting points where test is positive
LZONES=*linetype*	specifies line type for lines delineating zones A, B, and C
TESTFONT=*font*	specifies software font for labels at points where test is positive
TESTHEIGHT=*value*	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=*symbol*	specifies plot symbol for points where test is positive
TESTSYMBOLHT=*value*	specifies symbol height for points where test is positive
WTESTS=*n*	specifies width of line connecting points where test is positive

Table 45.5. Line Printer Options for Displaying Tests for Special Causes

TESTCHAR='*character*'	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
ZONECHAR='*character*'	specifies character for lines that delineate zones for tests for special causes

Table 45.6. Block Variable Legend Options

BLOCKLABELPOS=*keyword*	specifies position of label for *block-variable* legend
BLOCKLABTYPE=*n*|*keyword*	specifies text size of *block-variable* legend
BLOCKPOS=*n*	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=*color-list*	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=*variable*|(*variables*)	specifies one or more variables whose values are colors for filling background of *block-variable* legend

Table 45.7. Axis and Axis Label Options

CAXIS=*color*	specifies color for axis lines and tick marks
CFRAME=*color*|(*color-list*)	specifies fill colors for frame for plot area
CTEXT=*color*	specifies color for tick mark values and axis labels
HAXIS=*values*|AXISn	specifies major tick mark values for horizontal axis
HEIGHT=*value*	specifies height of axis label and axis legend text
HMINOR=*n*	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=*value*	specifies length of offset at both ends of horizontal axis
INTSTART=*value*	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	suppresses vertical axis truncation at zero applied by default
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=*n*	specifies thinning factor for tick mark labels on horizontal axis
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=*values*|AXISn	specifies major tick mark values for vertical axis
VFORMAT=*format*	specifies format for vertical axis tick mark labels
VMINOR=*n*	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=*value*	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis
WAXIS=*n*	specifies width of axis lines
YSCALE=PERCENT	scales vertical axis in percent units (rather than proportions)

Table 45.8. Options for Specifying Control Limits

ALPHA=*value*	requests probability limits for control charts
LIMITN=*n*|VARYING	specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	computes control limits for each *process* from the data rather than from a LIMITS= data set (Release 6.10 and later releases)
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEXES=ALL| '*label1*'...'*labeln*'	reads multiple sets of control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (Release 6.09 and earlier releases)
SIGMAS=*k*	specifies width of control limits in terms of multiple *k* of standard error of plotted proportion of nonconforming items

Table 45.9. Options for Displaying Control Limits

CINFILL=*color*	specifies color for area inside control limits
CLIMITS=*color*	specifies color of control limits, central line, and related labels
LCLLABEL='*label*'	specifies label for lower control limit
LIMLABSUBCHAR= '*character*'	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=*linetype*	specifies line type for control limits
NDECIMAL=*n*	specifies number of digits to right of decimal place in default labels for control limits and central line
NOCTL	suppresses display of central line
NOLCL	suppresses display of lower control limit
NOLIMITLABEL	suppresses labels for control limits and central line
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOLIMIT0	suppresses display of lower control limit if it is 0
NOLIMIT1	suppresses display of upper control limit if it is 1 (100%)
NOUCL	suppresses display of upper control limit
PSYMBOL='*string*' *keyword*	specifies label for central line
UCLLABEL='*string*'	specifies label for upper control limit
WLIMITS=*n*	specifies width for control limits and central line

Table 45.10. Grid Options

CGRID=*color*	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=*linetype*	specifies line type for grid requested with the ENDGRID option
LGRID=*linetype*	specifies line type for grid requested with the GRID option
WGRID=*n*	specifies width of grid lines

Table 45.11. Options for Plotting and Labeling Points

ALLLABEL=VALUE|(*variable*)	labels every point
CLABEL=*color*	specifies color for labels
CCONNECT=*color*	specifies color for line segments that connect points on chart
CFRAMELAB=*color*	specifies fill color for frame around labeled points
CNEEDLES=*color*	specifies color for needles that connect points to central line
CONNECTCHAR='*character*'	specifies character used to form line segments that connect points on chart
COUT=*color*	specifies color for portions of line segments that connect points outside control limits
COUTFILL=*color*	specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=*angle*	specifies angle at which labels are drawn
LABELFONT=*font*	specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=*value*	specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on chart
OUTLABEL=VALUE|(*variable*)	labels points outside control limits
SYMBOLCHARS='*characters*'	specifies characters indicating *symbol-variable*
SYMBOLLEGEND=NONE|*name*	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=*keyword*	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=*n*	specifies width of needles

Table 45.12. Options for Interactive Control Charts

HTML=(*variable*)	specifies a variable whose values are URLs to be associated with subgroups
HTML_LEGEND=(*variable*)	specifies a variable whose values are URLs to be associated with symbols in the symbol legend
WEBOUT=*SAS-data-set*	creates an OUTTABLE= data set with additional graphics coordinate data

Table 45.13. Clipping Options

CCLIP=*color*	specifies color for plot symbol for clipped points
CLIPCHAR='*character*'	specifies plot character for clipped points
CLIPFACTOR=*value*	determines extent to which extreme points are clipped
CLIPLEGEND='*string*'	specifies text for clipping legend
CLIPLEGPOS=*keyword*	specifies position of clipping legend
CLIPSUBCHAR='*character*'	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=*symbol*	specifies plot symbol for clipped points
CLIPSYMBOLHT=*value*	specifies symbol marker height for clipped points

Table 45.14. Phase Options

CPHASELEG=*color*	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE='*string*'	specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=*value*|*keyword*	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES= ALL|'*label1*'...'*labeln*'	specifies *phases* to be read from an input data set

Table 45.15. Standard Value Options

P0=*value*	specifies known (standard) value p_0 for proportion of nonconforming items p
TYPE=*keyword*	identifies whether parameters are estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set

Table 45.16. Input Data Set Options

DATAUNIT=*keyword*	specifies that input values are proportions or percentages (rather than counts) of nonconforming items
MISSBREAK	specifies that observations with missing values are not to be processed
SUBGROUPN=*n*| *variable*	specifies subgroup sample sizes as constant number n or as values of *variable* in a DATA= data set

Table 45.17. Output Data Set Options

OUTHISTORY= *SAS-data-set*	creates output data set containing subgroup proportions of nonconforming items and subgroup sample sizes
OUTINDEX='*string*'	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS= *SAS-data-set*	creates output data set containing control limits
OUTTABLE= *SAS-data-set*	creates output data set containing subgroup proportions of nonconforming items, subgroup sample sizes, and control limits

Table 45.18. Plot Layout Options

ALLN	plots proportion of nonconforming items for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts for a *process* only when exceptions occur
INTERVAL=*keyword*	specifies natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=*n*	specifies maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=*n*	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=*n*	specifies number of pages or screens to be used to display chart
ZEROSTD	displays p chart regardless of whether $\hat{\sigma} = 0$

Table 45.19. Graphical Enhancement Options

ANNOTATE=*SAS-data-set*	specifies annotate data set that adds features to chart
DESCRIPTION='*string*'	specifies string that appears in the description field of the PROC GREPLAY master menu
FONT=*font*	specifies software font for labels and legends on charts
NAME='*string*'	specifies name that appears in the name field of the PROC GREPLAY master menu
PAGENUM='*string*'	specifies the form of the label used in pagination
PAGENUMPOS=*keyword*	specifies the position of the page number requested with the PAGENUM= option

Table 45.20. Star Options

CSTARCIRCLES=*color*	specifies color for circles specified by the STARCIRCLES= option
CSTARFILL=*color*|*(variable)*	specifies color for filling stars
CSTAROUT=*color*	specifies outline color for stars exceeding inner or outer circles
CSTARS=*color*| *(variable)*	specifies color for outlines of stars
LSTARCIRCLES=*linetypes*	specifies line types for STARCIRCLES= circles
LSTARS=*linetype*|*(variable)*	specifies line types for outlines of stars requested with the STARVERTICES= option
STARBDRADIUS=*value*	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=*value-list*	specifies reference circles for stars
STARINRADIUS=*value*	specifies inner radius of stars
STARLABEL=*keyword*	specifies vertices to be labeled
STARLEGEND=*keyword*	specifies style of legend for star vertices
STARLEGENDLAB='*label*'	specifies label for STARLEGEND= legend
STAROUTRADIUS=*value*	specifies outer radius of stars
STARSPEC=*value*|*SAS-data-set*	specifies method used to standardize vertex variables
STARSTART=*value*	specifies angle for first vertex
STARTYPE=*keyword*	specifies graphical style of star
STARVERTICES=*variable*|*(variables)*	superimposes star at each point on chart
WSTARCIRCLES=*n*	specifies width of circles requested by the STARCIRCLES= option
WSTARS=*n*	specifies width of stars requested by the STARVERTICES= option

Table 45.21. Overlay Options

CCOVERLAY=*color-list*	specifies colors for overlay line segments
COVERLAY=*color-list*	specifies colors for overlay plots
COVERLAYCLIP=*color*	specifies color for clipped points on overlays
LOVERLAY=*linetypes*	specifies line types for overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=*variable-list*	specifies variables to overlay on control chart
OVERLAYCLIPSYM=*symbol*	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=*value*	specifies symbol height for clipped points on overlays
OVERLAYHTML=*variable-list*	specifies URLS to associate with overlay points
OVERLAYID=*variable-list*	specifies labels for overlay points
OVERLAYLEGLAB='*label*'	specifies label for overlay legend
OVERLAYSYM=*symbol-list*	specifies symbols for overlay plots
OVERLAYSYMHT=*value-list*	specifies symbol heights for overlay plots
WOVERLAY=*value-list*	specifies widths of overlay line segments

Details

Constructing Charts for Proportion Nonconforming (p Charts)

The following notation is used in this section:

p	expected proportion of nonconforming items produced by the process
p_i	proportion of nonconforming items in the i^{th} subgroup
X_i	number of nonconforming items in the i^{th} subgroup
n_i	number of items in the i^{th} subgroup
$\bar{p}$	average proportion of nonconforming items taken across subgroups: $$\bar{p} = \frac{n_1 p_1 + \cdots + n_N p_N}{n_1 + \cdots + n_N} = \frac{X_1 + \cdots + X_N}{n_1 + \cdots + n_N}$$
N	number of subgroups
$I_T(\alpha, \beta)$	incomplete beta function: $$I_T(\alpha,\beta) = (\Gamma(\alpha+\beta)/\Gamma(\alpha)\Gamma(\beta)) \int_0^T t^{\alpha-1}(1-t)^{\beta-1}dt$$ for $0 < T < 1$, $\alpha > 0$, and $\beta > 0$, where $\Gamma(\cdot)$ is the gamma function

Plotted Points

Each point on a p chart represents the observed proportion ($p_i = X_i/n_i$) of nonconforming items in a subgroup. For example, suppose the second subgroup (see Figure 45.10) contains 16 items, of which two are nonconforming. The point plotted for the second subgroup is $p_2 = 2/16 = 0.125$.

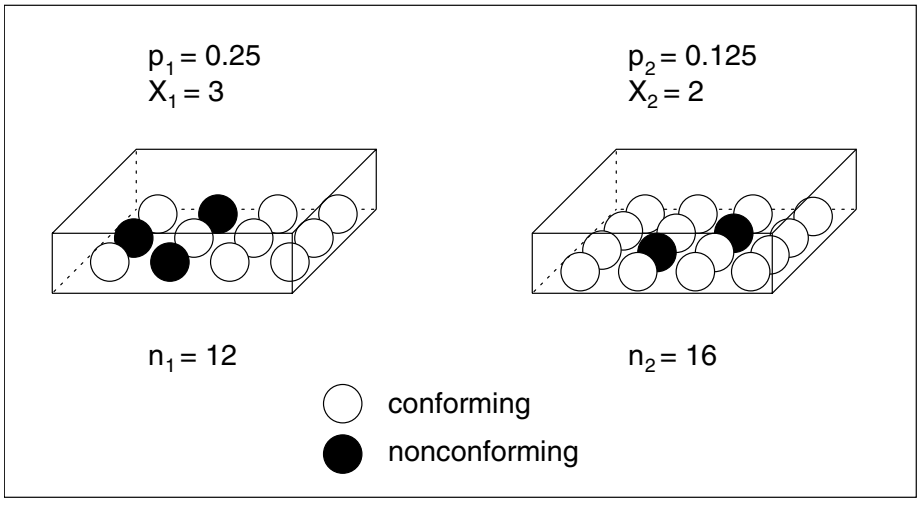


Figure 45.10. Proportions Versus Counts

Note that an np chart displays the number (count) of nonconforming items X_i. You can use the NPCHART statement to create np charts; see Chapter 44, "NPCHART Statement."

Central Line

By default, the central line on a p chart indicates an estimate of p that is computed as $\bar{p}$. If you specify a known value (p_0) for p, the central line indicates the value of p_0.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of p_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that p_i exceeds the limits

The lower and upper control limits, LCL and UCL, respectively, are computed as

$$\begin{aligned} \text{LCL} &= \max\left(\bar{p} - k\sqrt{\bar{p}(1-\bar{p})/n_i}\ , 0\right) \\ \text{UCL} &= \min\left(\bar{p} + k\sqrt{\bar{p}(1-\bar{p})/n_i}\ , 1\right) \end{aligned}$$

A lower probability limit for p_i can be determined using the fact that

$$\begin{aligned} P\{p_i < \text{LCL}\} &= 1 - P\{p_i \geq \text{LCL}\} \\ &= 1 - P\{X_i \geq n_i\text{LCL}\} \\ &= 1 - I_{\bar{p}}(n_i\text{LCL}, n_i + 1 - n_i\text{LCL}) \\ &= I_{1-\bar{p}}(n_i + 1 - n_i\text{LCL}, n_i\text{LCL}) \end{aligned}$$

Refer to Johnson, Kotz, and Kemp (1992). This assumes that the process is in statistical control and that X_i is binomially distributed. The lower probability limit LCL is then calculated by setting

$$I_{1-\bar{p}}(n_i + 1 - n_i\text{LCL}, n_i\text{LCL}) = \alpha/2$$

and solving for LCL. Similarly, the upper probability limit for p_i can be determined using the fact that

$$\begin{aligned} P\{p_i > \text{UCL}\} &= P\{p_i > \text{UCL}\} \\ &= P\{X_i > n_i\text{UCL}\} \\ &= I_{\bar{p}}(n_i\text{UCL}, n_i + 1 - n_i\text{UCL}) \end{aligned}$$

The upper probability limit UCL is then calculated by setting

$$I_{\bar{p}}(n_i\text{UCL}, n_i + 1 - n_i\text{UCL}) = \alpha/2$$

and solving for UCL. The probability limits are asymmetric around the central line. Note that both the control limits and probability limits vary with n_i.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify p_0 with the P0= option or with the variable _P_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables can be saved:

Table 45.22. OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLP	lower control limit for proportion of nonconforming items
LIMITN	nominal sample size associated with the control limits
_P_	average proportion of nonconforming items ($\bar{p}$ or p_0)
SIGMAS	multiple (k) of standard error of p_i
SUBGRP	*subgroup-variable* specified in the PCHART statement
TYPE	type (standard or estimate) of _P_
UCLP	upper control limit for proportion of nonconforming items
VAR	*process* specified in the PCHART statement

Notes:

1. If the control limits vary with subgroup sample size, the special missing value *V* is assigned to the variables _LIMITN_, _LCLP_, _UCLP_, and _SIGMAS_.
2. If the limits are defined in terms of a multiple k of the standard error of p_i, the value of _ALPHA_ is computed as $\alpha = P\{p_i < \text{\_LCLP\_}\} + P\{p_i > \text{\_UCLP\_}\}$, using the incomplete beta function.
3. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = (\text{\_UCLP\_} - \text{\_P\_})/\sqrt{\text{\_P\_}(1 - \text{\_P\_})/\text{\_LIMITN\_}}$. If _LIMITN_ has the special missing value *V*, this value is assigned to _SIGMAS_.
4. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the PCHART statement. For an example, see "Saving Control Limits" on page 1534.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables are saved:

- the *subgroup-variable*
- a subgroup proportion of nonconforming items variable named by *process* suffixed with *P*
- a subgroup sample size variable named by *process* suffixed with *N*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the PCHART statement. For example, consider the following statements:

```
proc shewhart data=input;
   pchart (rework rejected)*batch / outhistory=summary
                                    subgroupn =30;
run;
```

The data set SUMMARY contains variables named BATCH, REWORKP, REWORKN, REJETEDP, and REJETEDN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Proportions of Nonconforming Items" on page 1533.

Note that an OUTHISTORY= data set created with the PCHART statement can be reused as a HISTORY= data set by either the PCHART statement or the NPCHART statement.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The variables shown in the following table are saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on p chart
LCLP	lower control limit for proportion of nonconforming items
LIMITN	nominal sample size associated with the control limits
SIGMAS	multiple (k) of the standard error of p_i associated with the control limits
subgroup	values of the subgroup variable
SUBP	subgroup proportion of nonconforming items
SUBN	subgroup sample size
TESTS	tests for special causes signaled on p chart
UCLP	upper control limit for proportion of nonconforming items
VAR	*process* specified in the PCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The k^{th} character of a value of _TESTS_ is k if Test k is positive at that subgroup. For example, if you request the first four tests (the tests appropriate for p charts) and Tests 2 and 4 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, a 4 for the fourth character, and blanks for the other six characters.
3. The variables _EXLIM_ and _TESTS_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1534.

ODS Tables

The following table summarizes the ODS tables that you can request with the PCHART statement.

Table 45.23. ODS Tables Produced with the PCHART Statement

Table Name	Description	Options
PCHART	*p* chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

Input Data Sets

DATA= Data Set

You can read raw data (counts of nonconforming items) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the PCHART statement must be a SAS variable in the DATA= data set. This variable provides counts for subgroup samples indexed by the values of the *subgroup-variable*. The *subgroup-variable*, which is specified in the PCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a count for each *process* and a value for the *subgroup-variable*. The data set must contain one observation for each subgroup. Note that you can specify the DATAUNIT= option in the PCHART statement to read proportions or percentages of nonconforming items instead of counts. Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

When you use a DATA= data set with the PCHART statement, the SUBGROUPN= option (which specifies the subgroup sample size) is required. By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 1936).

For an example of a DATA= data set, see "Creating p Charts from Count Data" on page 1528.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set CONLIMS:*

```
proc shewhart data=info limits=conlims;
   pchart rejects*batch / subgroupn= 100;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLP_, _P_, and _UCLP_, which specify the control limits directly
- the variable _P_, without providing _LCLP_ and _UCLP_. The value of _P_ is used to calculate the control limits according to the equations on page 1550.

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are `ESTIMATE` and `STANDARD`.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1536.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This allows you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART procedure or to create your own HISTORY= data set.

A HISTORY= data set used with the PCHART statement must contain the following:

- the *subgroup-variable*
- a subgroup proportion of nonconforming items variable for each *process*
- a subgroup sample size variable for each *process*

*In Release 6.09 and in earlier releases, it is necessary to specify the READLIMITS option.

The names of the proportion sample size variables must be the *process* name concatenated with the special suffix characters P and N, respectively.

For example, consider the following statements:

```
proc shewhart history=summary;
   pchart (rework rejected)*batch / subgroupn=50;
run;
```

The data set SUMMARY must include the variables BATCH, REWORKP, REWORKN, REJETEDP, and REJETEDN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 1936 for an example).

For an example of a HISTORY= data set, see "Creating p Charts from Summary Data" on page 1530.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information read from a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in Chapter 56, "Specialized Control Charts."

The following table lists the variables required in a TABLE= data set used with the PCHART statement:

Table 45.24. Variables Required in a TABLE= Data Set

Variable	Description
LCLP	lower control limit for proportion of nonconforming items
LIMITN	nominal sample size associated with the control limits
_P_	average proportion of nonconforming items
subgroup-variable	values of the *subgroup-variable*
SUBN	subgroup sample size
SUBP	subgroup proportion of nonconforming items
UCLP	upper control limit for proportion of nonconforming items

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1534.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical	DATA=	*process*
Vertical	HISTORY=	subgroup proportion nonconforming variable
Vertical	TABLE=	_SUBP_

For example, the following sets of statements specify the label *Proportion Nonconforming* for the vertical axis of the *p* chart:

```
proc shewhart data=circuits;
   pchart fail*batch / subgroupn=500;
   label fail = 'Proportion Nonconforming';
run;

proc shewhart history=cirhist;
   pchart fail*batch ;
   label failp = 'Proportion Nonconforming';
run;

proc shewhart table=cirtable;
   pchart fail*batch ;
   label _SUBP_ = 'Proportion Nonconforming';
run;
```

In this example, the label assignments are in effect only for the duration of the procedure step, and they temporarily override any permanent labels associated with the variables.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples

This section provides advanced examples of the PCHART statement.

Example 45.1. Applying Tests for Special Causes

See SHWPEX1 in the SAS/QC Sample Library

This example shows how you can apply tests for special causes to make p charts more sensitive to special causes of variation. The following statements create a SAS data set named CIRCUIT3, which contains the number of failing circuits for 20 batches from the circuit manufacturing process introduced in "Creating p Charts from Count Data" on page 1528:

```
data circuit3;
   input batch fail @@;
datalines;
 1 12    2  21    3  16    4  9
 5  3    6   4    7   6    8  9
 9 11   10  13   11  12   12  7
13  2   14  14   15   9   16  8
17 14   18  10   19  11   20  9
;
run;
```

The following statements create the p chart, apply several tests to the chart, and tabulate the results:

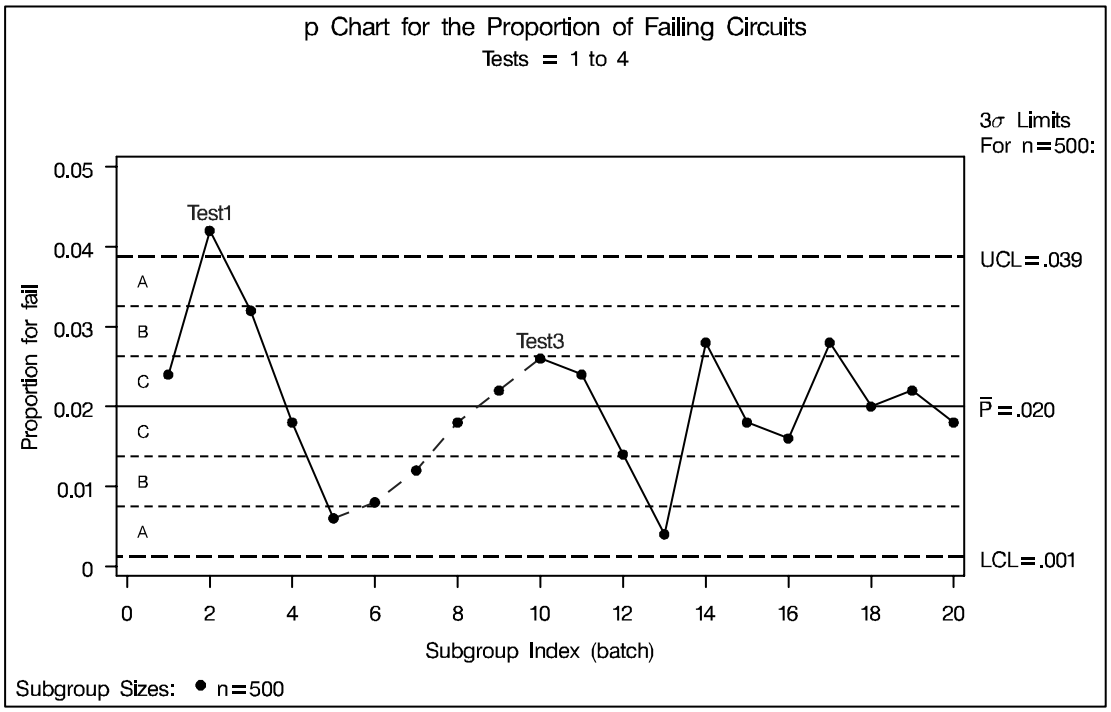
```
title1'p Chart for the Proportion of Failing Circuits';
title2 'Tests = 1 to 4';
proc shewhart data=circuit3;
   pchart fail*batch / subgroupn = 500
                       tests     = 1 to 4
                       zones
                       zonelabels
                       ltests    = 20
                       table
                       tabletest
                       tablelegend;
run;
```

The chart is shown in Output 45.1.1, and the printed output is shown in Output 45.1.2. The TESTS= option requests Tests 1, 2, 3, and 4, which are described in Chapter 55, "Tests for Special Causes." The TABLETESTS option requests a table of proportions of nonconforming items and control limits, with a column indicating which subgroups tested positive for special causes. The TABLELEGEND option adds a legend describing the tests that are positive.

The ZONELABELS option displays zone lines and zone labels on the chart. The zones are used to define the tests. The LTESTS= option specifies the line type used to connect the points in a pattern for a test that is signaled.

Output 45.1.1 and Output 45.1.2 indicate that Test 1 is positive at batch 2 and Test 3 is positive at batch 10.

Output 45.1.1. Tests for Special Causes Displayed on p Chart



Output 45.1.2. Tabular Form of p Chart

p Chart for the Proportion of Failing Circuits
Tests = 1 to 4

p Chart Summary for fail

batch	Subgroup Sample Size	-3 Sigma Limits with n=500 for Proportion- Lower Limit	Subgroup Proportion	Upper Limit	Special Tests Signaled
1	500	0.00121703	0.02400000	0.03878297	
2	500	0.00121703	0.04200000	0.03878297	1
3	500	0.00121703	0.03200000	0.03878297	
4	500	0.00121703	0.01800000	0.03878297	
5	500	0.00121703	0.00600000	0.03878297	
6	500	0.00121703	0.00800000	0.03878297	
7	500	0.00121703	0.01200000	0.03878297	
8	500	0.00121703	0.01800000	0.03878297	
9	500	0.00121703	0.02200000	0.03878297	
10	500	0.00121703	0.02600000	0.03878297	3
11	500	0.00121703	0.02400000	0.03878297	
12	500	0.00121703	0.01400000	0.03878297	
13	500	0.00121703	0.00400000	0.03878297	
14	500	0.00121703	0.02800000	0.03878297	
15	500	0.00121703	0.01800000	0.03878297	
16	500	0.00121703	0.01600000	0.03878297	
17	500	0.00121703	0.02800000	0.03878297	
18	500	0.00121703	0.02000000	0.03878297	
19	500	0.00121703	0.02200000	0.03878297	
20	500	0.00121703	0.01800000	0.03878297	

Test Descriptions

Test 1 One point beyond Zone A (outside control limits)
Test 3 Six points in a row steadily increasing or decreasing

Example 45.2. Specifying Standard Average Proportion

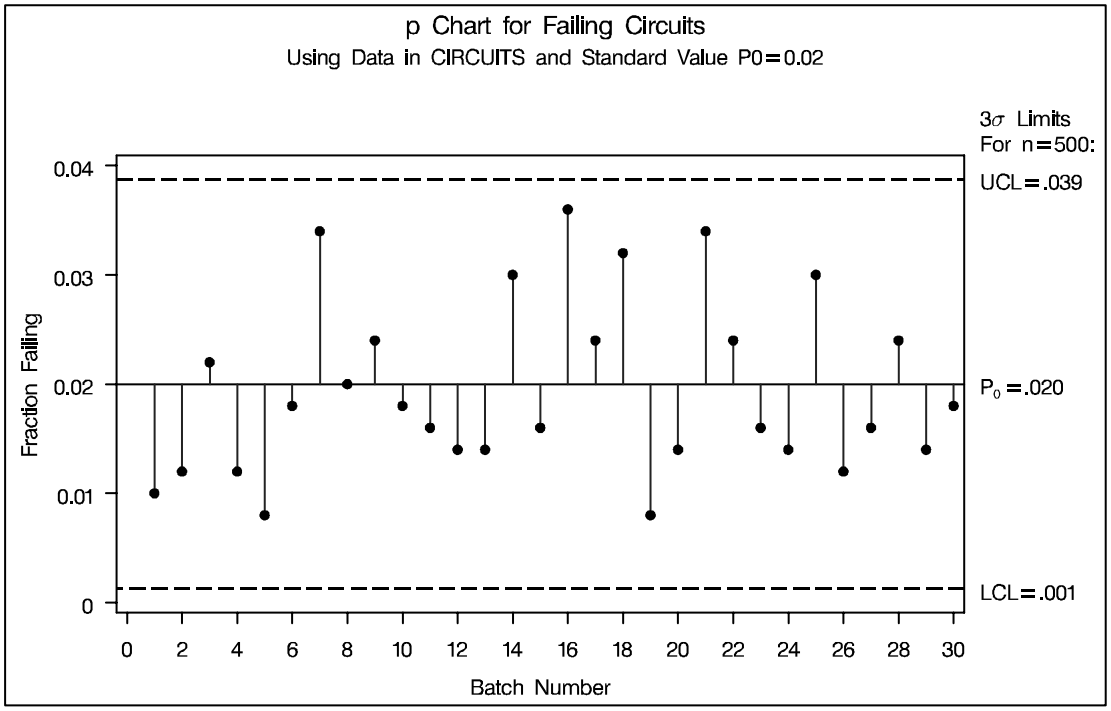
See SHWPEX2 in the SAS/QC Sample Library

In some situations, a standard (known) value (p_0) is available for the expected proportion of nonconforming items, based on extensive testing or previous sampling. This example illustrates how you can specify p_0 to create a p chart.

A p chart is used to monitor the proportion of failing circuits in the data set CIRCUITS, which is introduced on page 1528. The expected proportion is known to be $p_0 = 0.02$. The following statements create a p chart, shown in Output 45.2.1, using p_0 to compute the control limits:

```
title1 'p Chart for Failing Circuits';
title2 'Using Data in CIRCUITS and Standard Value P0=0.02';
proc shewhart data=circuits;
   pchart fail*batch / subgroupn = 500
                       p0        = 0.02
                       psymbol   = p0
                       needles
                       nolegend;
   label batch = 'Batch Number'
         fail  = 'Fraction Failing';
run;
```

Output 45.2.1. A p Chart with Standard Value p_0



The chart indicates that the process is in control. The P0= option specifies p_0. The PSYMBOL= option specifies a label for the central line indicating that the line represents a standard value. The NEEDLES option connects points to the central line with vertical needles. The NOLEGEND option suppresses the default legend for subgroup sample sizes. Labels for the vertical and horizontal axes are provided with the LABEL statement. For details concerning axis labeling, see "Axis Labels" on page 1557.

Alternatively, you can specify p_0 using the variable _P_ in a LIMITS= data set,* as follows:

```
data climits;
   length _var_ _subgrp_ _type_ $8;
   _p_      = 0.02;
   _subgrp_ = 'batch';
   _var_    = 'fail';
   _type_   = 'STANDARD';
   _limitn_ = 500;
run;
```

*In Release 6.09 and in earlier releases, it is also necessary to specify the READLIMITS option

```
proc shewhart data=circuits limits=climits;
   pchart fail*batch / subgroupn = 500
                       psymbol   = p0
                       nolegend
                       needles;
   label batch ='Batch Number'
         fail  ='Fraction Failing';
run;
```

The bookkeeping variable _TYPE_ indicates that _P_ has a standard value. The chart produced by these statements is identical to the chart in Output 45.2.1.

Example 45.3. Working with Unequal Subgroup Sample Sizes

See SHWPEX3 in the SAS/QC Sample Library

The following statements create a SAS data set named BATTERY, which contains the number of alkaline batteries per lot failing an acceptance test. The number of batteries tested in each lot varies but is approximately 150.

```
data battery;
   length lot $3;
   input lot nfailed sampsize @@;
   label nfailed  = 'Number Failed'
         lot      = 'Lot Number'
         sampsize = 'Number Sampled';
   datalines;
AE3  6  151     AE4  5  142     AE9  6  145
BR3  9  149     BR7  3  150     BR8  0  156
BR9  4  150     DB1  9  158     DB2  4  152
DB3  0  162     DB5  9  140     DB6  7  161
DS4  6  154     DS6  1  144     DS8  5  154
JG1  3  151     MC3  8  148     MC4  2  143
MK6  4  150     MM1  4  147     MM2  0  150
RT5  2  154     RT9  8  149     SP1  3  160
SP3  9  153
;
run;
```

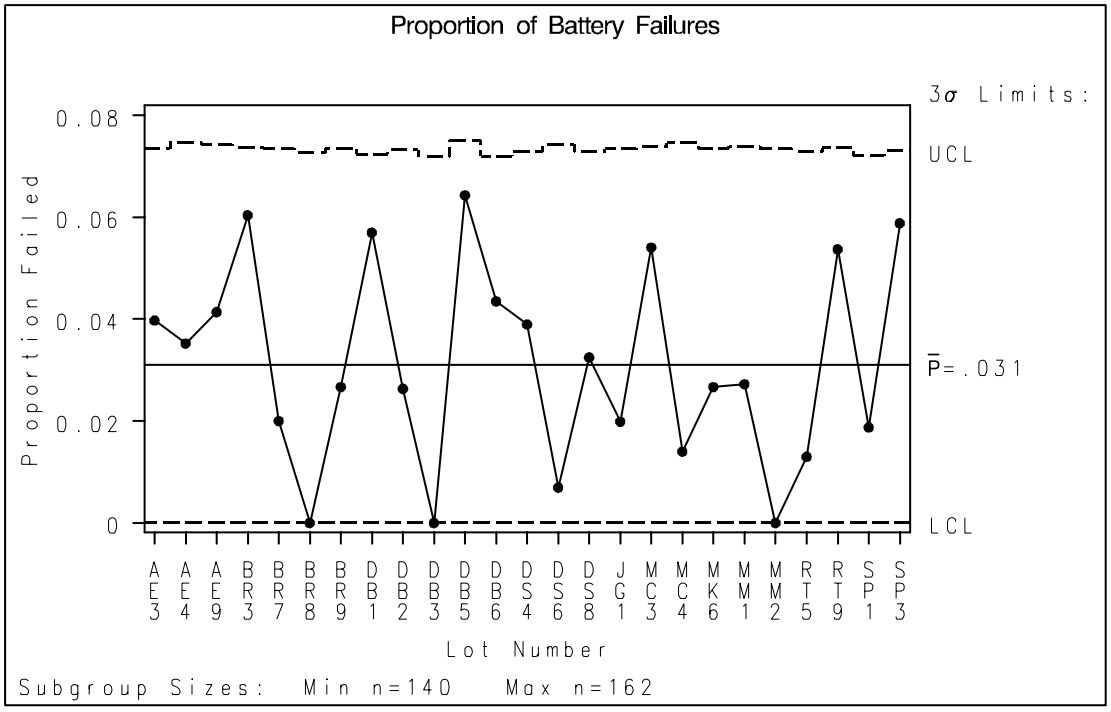
The variable NFAILED contains the number of battery failures, the variable LOT contains the lot number, and the variable SAMPSIZE contains the lot sample size. The following statements request a p chart for this data:

```
title 'Proportion of Battery Failures';
proc shewhart data=battery;
   pchart nfailed*lot / subgroupn = sampsize
                        turnhlabels
                        font      = 'Lucida Console'
                        outlimits = batlim;
   label nfailed = 'Proportion Failed';
run;
```

Here the FONT= option is used to specify the name of a hardware font to be used for the p chart. In this case the requested font is Lucida Console, a Windows TrueType font. See *SAS/GRAPH Software: Reference* and *SAS Companion for Microsoft Windows* for more information on hardware and TrueType fonts.

The chart is shown in Output 45.3.1 and the OUTLIMITS= data set BATLIM is listed in Output 45.3.2.

Output 45.3.1. A p Chart with Varying Subgroup Sample Sizes



Note that the upper control limit varies with the subgroup sample size. The lower control limit is truncated at zero. The sample size legend indicates the minimum and maximum subgroup sample sizes.

Output 45.3.2. Listing of the Control Limits Data Set BATLIM

Control Limits for Battery Failures

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLP_	_P_	_UCLP_
nfailed	lot	ESTIMATE	V	V	3	V	0.031010	V

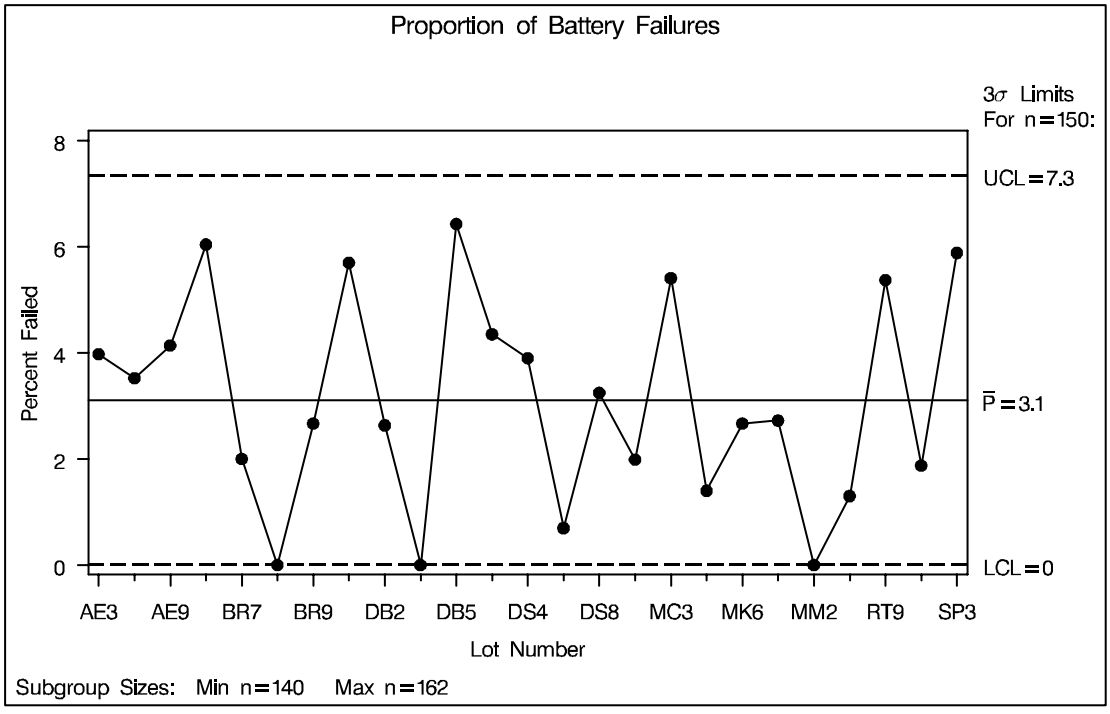
The variables in BATLIM whose values vary with subgroup sample size are assigned the special missing value *V*.

The SHEWHART procedure provides various options for working with unequal subgroup sample sizes. For example, you can use the LIMITN= option to specify a fixed (nominal) sample size for computing the control limits, as illustrated by the following statements:

```
title 'Proportion of Battery Failures';
proc shewhart data=battery;
   pchart nfailed*lot / subgroupn = sampsize
                        limitn    = 150
                        alln
                        outlimits = clim2
                        yscale    = percent;
   label nfailed = 'Percent Failed';
run;
```

The ALLN option specifies that all points (regardless of subgroup sample size) are to be displayed. By default, only points for subgroups whose sample size matches the LIMITN= value are displayed. The YSCALE= option specifies that the vertical axis is to be scaled in percentages rather than proportions. The chart is shown in Output 45.3.3.

Output 45.3.3. Control Limits Based on Fixed Subgroup Sample Size



All the points are inside the control limits, indicating that the process is in statistical control. Since there is relatively little variation in the sample sizes, the control limits in Output 45.3.3 provide a close approximation to the exact control limits in Output 45.3.1, and the same conclusions can be drawn from both charts. In general, care should be taken when interpreting charts that use a nominal sample size to compute control limits, since these limits are only approximate when the sample sizes vary.

Example 45.4. Creating a Chart with Revised Control Limits

The following statements create a SAS data set named CIRC_ONE, which contains the number of failing circuits for 30 batches produced by the circuit manufacturing process introduced in the "Getting Started" section on page 1528:

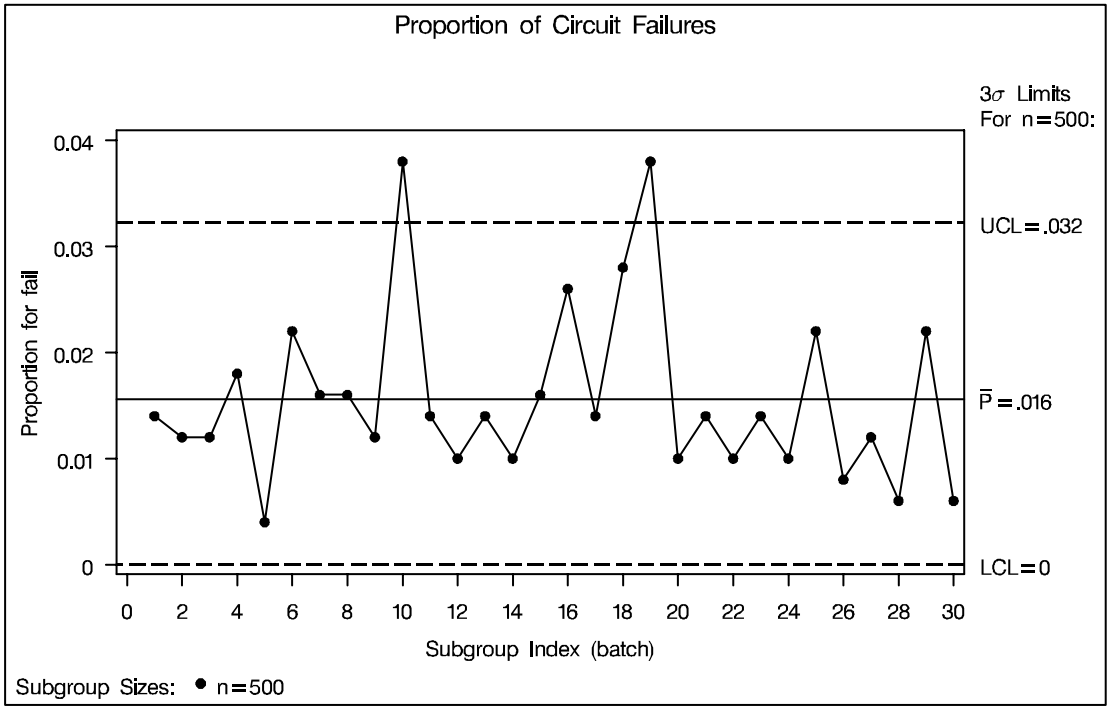
See SHWPEX4 in the SAS/QC Sample Library

```
data circ_one;
  input batch fail @@;
  datalines;
 1   7   2   6   3   6   4   9   5   2
 6  11   7   8   8   8   9   6  10  19
11   7  12   5  13   7  14   5  15   8
16  13  17   7  18  14  19  19  20   5
21   7  22   5  23   7  24   5  25  11
26   4  27   6  28   3  29  11  30   3
;
run;
```

A p chart is used to monitor the proportion of failing circuits. The following statements create the chart shown in Output 45.4.1:

```
title 'Proportion of Circuit Failures';
proc shewhart data=circ_one;
   pchart fail*batch / subgroupn = 500
                       outindex  = 'Trial Limits'
                       outlimits = faillim1;
run;
```

Output 45.4.1. A p Chart for Circuit Failures



Batches 10 and 19 have unusually high proportions of failing circuits. Subsequent investigation identifies special causes for both batches, and it is decided to eliminate these batches from the data set and recompute the control limits. The following statements create a data set named FAILLIM2 that contains the revised control limits:

```
proc shewhart data=circ_one;
   where batch ^= 10 and batch ^= 19;
   pchart fail*batch / subgroupn = 500
                       nochart
                       outindex  ='Revised Limits'
                       outlimits = faillim2;
run;

data faillims;
   set faillim1 faillim2;
run;
```

The data set FAILLIMS, which contains the true and revised control limits, is listed in Output 45.4.2.

Output 45.4.2. Listing of the Data Set FAILLIMS

```
                        Proportion of Circuit Failures


         _                    _           _
         S          _         L           _ S
         U          I     _   I           A I _         _
   _     B          N     T   M           L G L         U
   V     G          D     Y   I           P M C         C
   A     R          E     P   T           H A L   _     L
   R     P          X     E   N           A S P   P     P

   _     _          _     _   _           _ _ _   _     _

 fail batch Trial Limits   ESTIMATE 500 .005620297 3 0 0.0156 0.032226
 fail batch Revised Limits ESTIMATE 500 .005942336 3 0 0.0140 0.029763
```

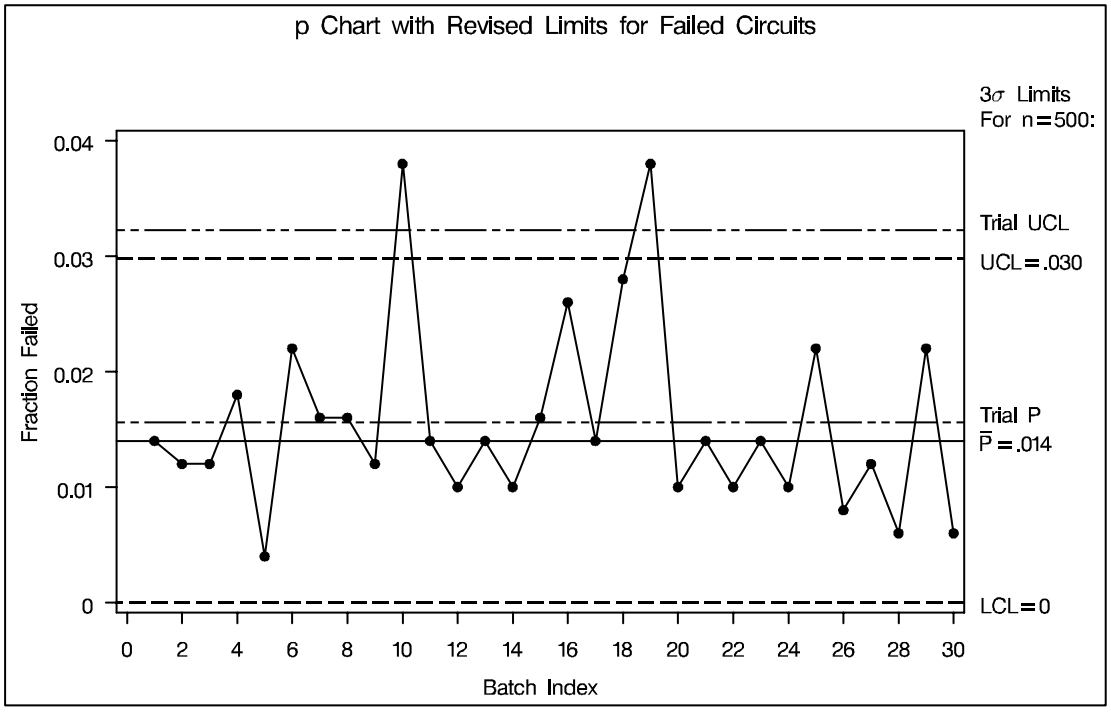
The following statements create a p chart displaying both sets of control limits:

```
title 'p Chart with Revised Limits for Failed Circuits';
proc shewhart data=circ_one limits=faillims;
   pchart fail*batch / subgroupn  = 500
                       readindex  = 'Revised Limits'
                       vref       = 0.0156 0.032226
                       vreflabels = ('Trial P' 'Trial UCL')
                       vreflabpos = 3
                       lvref      = 15
                       nolegend;
   label fail  = 'Fraction Failed'
         batch = 'Batch Index';
run;
```

The READINDEX= option is used to select the revised limits displayed on the p chart in Output 45.4.3. See “Displaying Multiple Sets of Control Limits” on page 1939.

The VREF=, VREFLABELS=, and VREFLABPOS= options are used to display and label the trial limits. You can also pass in the values of the trial limits with macro variables. For an illustration of this technique, see Example 39.6 on page 1295.

Output 45.4.3. p Chart with Revised Limits



Example 45.5. OC Curve for Chart

See SHWPOC in the SAS/QC Sample Library

This example uses the GPLOT procedure and the OUTLIMITS= data set FAILLIM2 from the previous example to plot an OC curve for the p chart shown in Output 45.4.3.

The OC curve displays β (the probability that p_i lies within the control limits) as a function of p (the true proportion nonconforming). The computations are exact, assuming that the process is in control and that the number of nonconforming items (X_i) has a binomial distribution.

The value of β is computed as follows:

$$\begin{aligned}
\beta &= P(p_i \le \text{UCL}) - P(p_i < \text{LCL}) \\
&= P(X_i \le n\text{UCL}) - P(X_i < n\text{LCL}) \\
&= P(X_i < n\text{UCL}) + P(X_i = n\text{UCL}) - P(X_i < n\text{LCL}) \\
&= I_{1-p}(n+1-n\text{UCL}, n\text{UCL}) + P(X_i = n\text{UCL}) - I_{1-p}(n+1-n\text{LCL}, n\text{LCL}) \\
&= I_p(n\text{LCL}, n+1-n\text{LCL}) + P(X_i = n\text{UCL}) - I_p(n\text{UCL}, n+1-n\text{UCL})
\end{aligned}$$

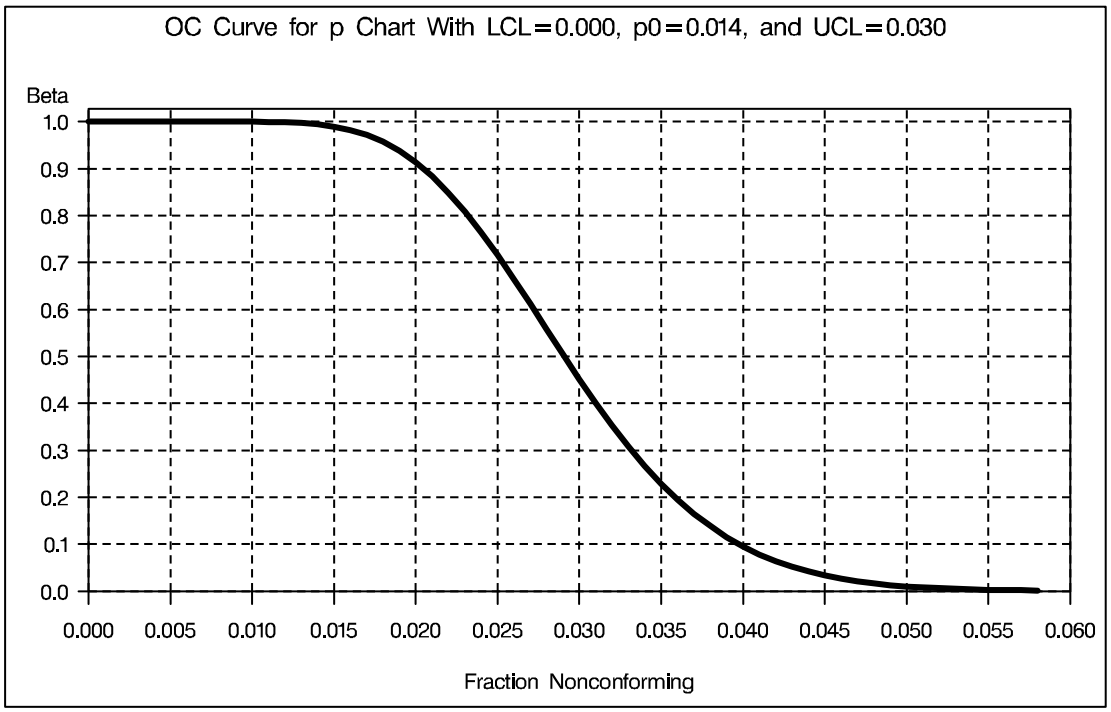
Here, $I_p(\cdot,\cdot)$ denotes the incomplete beta function. The following DATA step computes β (the variable BETA) as a function of p (the variable P):

```
data ocpchart;
   set faillim2;
   keep beta fraction _lclp_ _p_ _uclp_;
   nucl=_limitn_*_uclp_;
   nlcl=_limitn_*_lclp_;
   do p=0 to 500;
      fraction=p/1000;
      if nucl=floor(nucl) then
         adjust=probbnml(fraction,_limitn_,nucl) -
                probbnml(fraction,_limitn_,nucl-1);
      else adjust=0;
      if nlcl=0 then
         beta=1 - probbeta(fraction,nucl,_limitn_-nucl+1) + adjust;
      else beta=probbeta(fraction,nlcl,_limitn_-nlcl+1) -
                probbeta(fraction,nucl,_limitn_-nucl+1) +
                adjust;
      if beta >= 0.001 then output;
      end;
   call symput('lcl', put(_lclp_,5.3));
   call symput('mean',put(_p_,   5.3));
   call symput('ucl', put(_uclp_,5.3));
run;
```

The following statements display the OC curve shown in Output 45.5.1:

```
axis1 offset=(0,.5) minor=none order=0 to 1.0 by 0.1;
axis2 offset=(0,0)  minor=none order=0 to 0.06 by 0.005;

symbol i = j w = 3 v = none;
title "OC Curve for p Chart With LCL=&LCL, p0=&MEAN, and UCL=&UCL";
proc gplot data=ocpchart;
   plot beta*fraction /
      vaxis=axis1
      haxis=axis2
      frame
      autovref
      autohref
      lvref = 2
      lhref = 2
      vzero
      hzero;
   label fraction = 'Fraction Nonconforming'
         beta     = 'Beta';
run;
```

Output 45.5.1. OC Curve for p Chart



Chapter 46
RCHART Statement

Chapter Contents

OVERVIEW 1573

GETTING STARTED 1574
- Creating Range Charts from Raw Data 1574
- Creating Range Charts from Summary Data 1576
- Saving Summary Statistics 1579
- Saving Control Limits 1580
- Reading Preestablished Control Limits 1583

SYNTAX 1585
- Summary of Options 1586

DETAILS 1595
- Constructing Range Charts 1595
- Output Data Sets 1596
- ODS Tables 1600
- Input Data Sets 1600
- Methods for Estimating the Standard Deviation 1603
- Axis Labels 1604
- Missing Values 1604

EXAMPLES 1605
- Example 46.1. Computing Probability Limits 1605
- Example 46.2. Specifying Control Limit Information 1607

Chapter 46
RCHART Statement

Overview

The RCHART statement creates an R chart for subgroup ranges, which is used to analyze the variability of a process.*

You can use options in the RCHART statement to

- compute control limits from the data based on a multiple of the standard error of the plotted ranges or as probability limits
- tabulate subgroup sample sizes, subgroup ranges, control limits, and other information
- save control limits in an output data set
- save subgroup sample sizes, subgroup means, and subgroup ranges in an output data set
- read preestablished control limits from a data set
- specify the method for estimating the process standard deviation
- specify a known (standard) process standard deviation for computing control limits
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

*You can also use s charts for this purpose; see Chapter 47, "SCHART Statement." In general, s charts are recommended with large subgroup sample sizes ($n_i \geq 10$).

Getting Started

This section introduces the RCHART statement with simple examples that illustrate the most commonly used options. Complete syntax for the RCHART statement is presented in the "Syntax" section on page 1585, and advanced examples are given in the "Examples" section on page 1605.

Creating Range Charts from Raw Data

See SHWRCHR in the SAS/QC Sample Library

A disk drive manufacturer performs a battery of tests to evaluate its drives. The following statements create a data set named DISKS, which contains the time (in milliseconds) required to complete one of these tests for six drives in each of 25 lots:

```
data disks;
   input lot @;
   do i=1 to 6;
      input time @;
      output;
      end;
   drop i;
  datalines;
  1 8.05 7.90 8.04 8.06 8.01 7.99
  2 8.03 8.06 8.02 8.02 7.97 8.03
  3 8.00 7.94 7.97 7.95 8.00 8.01
  4 8.00 8.06 8.06 7.99 7.97 7.96
  5 7.93 8.01 8.00 8.09 8.06 8.02
  6 7.98 7.99 8.01 8.09 8.00 7.97
  7 8.00 7.94 7.93 8.03 7.93 8.08
  8 8.01 7.98 7.98 8.07 8.05 8.09
  9 7.97 7.96 8.01 8.11 8.06 8.07
 10 7.93 8.03 8.03 8.00 7.93 8.03
 11 8.00 8.00 8.02 7.92 7.98 8.01
 12 7.98 7.93 8.01 7.97 8.02 8.00
 13 8.06 7.93 7.98 7.98 8.02 7.96
 14 8.05 7.98 8.05 7.99 7.95 7.99
 15 7.94 8.01 7.97 8.04 7.91 8.03
 16 8.03 8.03 8.02 8.06 8.00 7.97
 17 8.03 7.94 8.05 8.05 8.04 7.94
 18 7.99 7.99 7.86 7.99 8.06 8.03
 19 7.95 7.96 7.99 7.96 7.94 8.12
 20 8.03 8.07 7.98 7.97 8.00 8.04
 21 8.04 7.90 8.03 8.02 7.98 7.97
 22 7.95 8.05 7.98 8.01 7.97 8.15
 23 8.06 8.00 8.03 8.02 7.99 7.95
 24 7.97 8.02 8.00 7.96 7.96 8.00
 25 8.12 7.97 7.99 8.09 8.05 8.00
;
run;
```

A partial listing of DISKS is shown in Figure 46.1.

The Data Set DISKS

lot	time
1	8.05
1	7.90
1	8.04
1	8.06
1	8.01
1	7.99
2	8.03
2	8.06
2	8.02
2	8.02
2	7.97
2	8.03
3	8.00
3	7.94
3	7.97
3	7.95
3	8.00
3	8.01
.	.
.	.
.	.

Figure 46.1. Partial Listing of the Data Set DISKS

The data set DISKS is said to be in "strung-out" form since each observation contains the lot number and test time for a single disk drive. The first five observations contain the times for the first lot, the second five observations contain the times for the second lot, and so on. Because the variable LOT classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable TIME contains the time measurements and is referred to as the *process variable* (or *process* for short).

You can use an R chart to determine whether the variability in the performance of the disk drives is in control. The following statements create the R chart shown in Figure 46.2:

```
title 'Range Chart for Disk Drive Test Times';
proc shewhart data=disks;
   rchart time*lot;
run;
```

This example illustrates the basic form of the RCHART statement. After the keyword RCHART, you specify the *process* to analyze (in this case, TIME), followed by an asterisk and the *subgroup-variable* (LOT).

The input data set is specified with the DATA= option in the PROC SHEWHART statement.

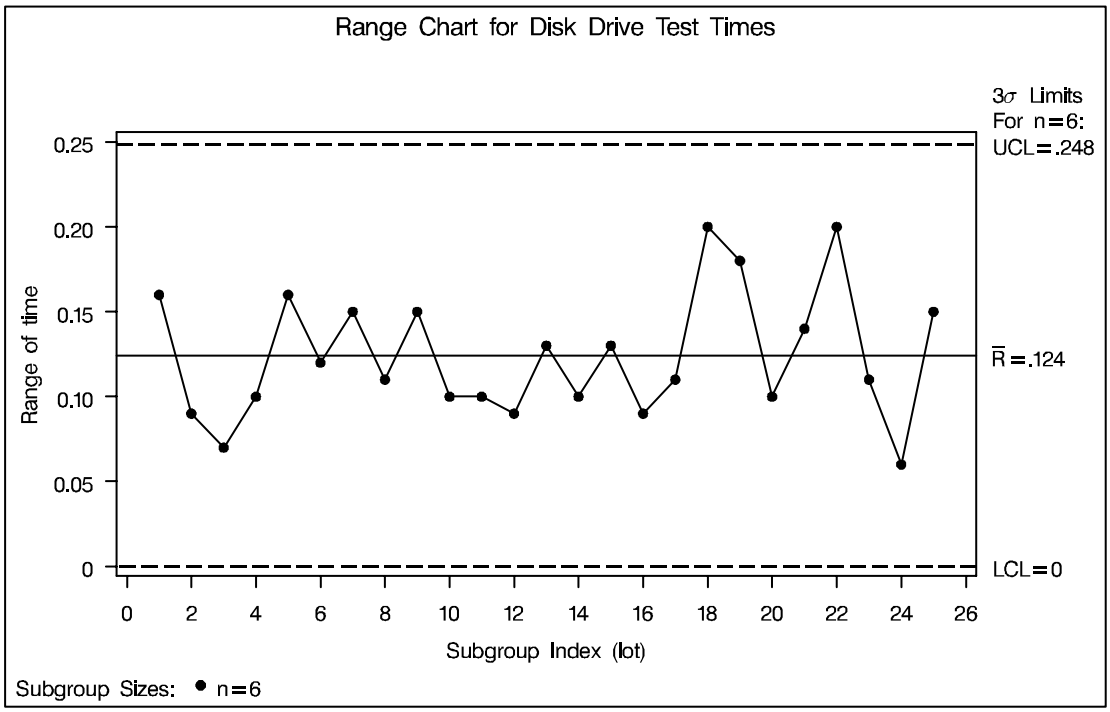


Figure 46.2. R Chart for the Data Set DISKS

Each point on the R chart represents the range of the measurements for a particular lot. For instance, the range plotted for the first lot is $8.06 - 7.90 = 0.16$. Since all of the subgroup ranges lie within the control limits, you can conclude that the variability in the performance of the disk drives is in statistical control.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 46.23 on page 1596. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1583.

For computational details, see "Constructing Range Charts" on page 1595. For more details on reading raw data, see "DATA= Data Set" on page 1600.

Creating Range Charts from Summary Data

See SHWRCHR in the SAS/QC Sample Library

The previous example illustrates how you can create R charts using raw data (process measurements). However, in many applications the data are provided as subgroup summary statistics. This example illustrates how you can use the RCHART statement with data of this type.

The following data set (DISKSUM) provides the data from the preceding example in summarized form:

```
data disksum;
   input lot timex timer;
   timen=6;
datalines;
  1  8.00833  0.16
  2  8.02167  0.09
  3  7.97833  0.07
  4  8.00667  0.10
  5  8.01833  0.16
  6  8.00667  0.12
  7  7.98500  0.15
  8  8.03000  0.11
  9  8.03000  0.15
 10  7.99167  0.10
 11  7.98833  0.10
 12  7.98500  0.09
 13  7.98833  0.13
 14  8.00167  0.10
 15  7.98333  0.13
 16  8.01833  0.09
 17  8.00833  0.11
 18  7.98667  0.20
 19  7.98667  0.18
 20  8.01500  0.10
 21  7.99000  0.14
 22  8.01833  0.20
 23  8.00833  0.11
 24  7.98500  0.06
 25  8.03667  0.15
;
run;
```

A partial listing of DISKSUM is shown in Figure 46.3. There is exactly one observation for each subgroup (note that the subgroups are still indexed by LOT). The variable TIMEX contains the subgroup means, the variable TIMER contains the subgroup ranges, and the variable TIMEN contains the subgroup sample sizes (these are all six).

The Summary Data Set of Disk Drive Test Times

lot	timex	timer	timen
1	8.00833	0.16	6
2	8.02167	0.09	6
3	7.97833	0.07	6
4	8.00667	0.10	6
5	8.01833	0.16	6
.	.	.	.
.	.	.	.
.	.	.	.

Figure 46.3. The Summary Data Set DISKSUM

You can read this data set by specifying it as a HISTORY= data set in the PROC SHEWHART statement, as follows:

```
options ls=88 ps=30;
title 'Range Chart for Disk Drive Test Times';
proc shewhart history=disksum lineprinter;
   rchart time*lot='X';
run;
options ls=76 ps=80;
```

The resulting R chart is shown in Figure 46.4. Since the LINEPRINTER option is specified in the PROC SHEWHART statement, line printer output is produced. The character (X) specified in single quotes after the *subgroup-variable* specifies the *character* used to plot points. This character must follow an equal sign.

Note that TIME is *not* the name of a SAS variable in the data set DISKSUM but is, instead, the common prefix for the names of the SAS variables TIMER and TIMEN. The suffix characters R and N indicate *range* and *sample size*, respectively. Thus, you can specify two subgroup summary variables in the HISTORY= data set with a single name (TIME), which is referred to as the *process*. The name LOT specified after the asterisk is the name of the *subgroup-variable*.

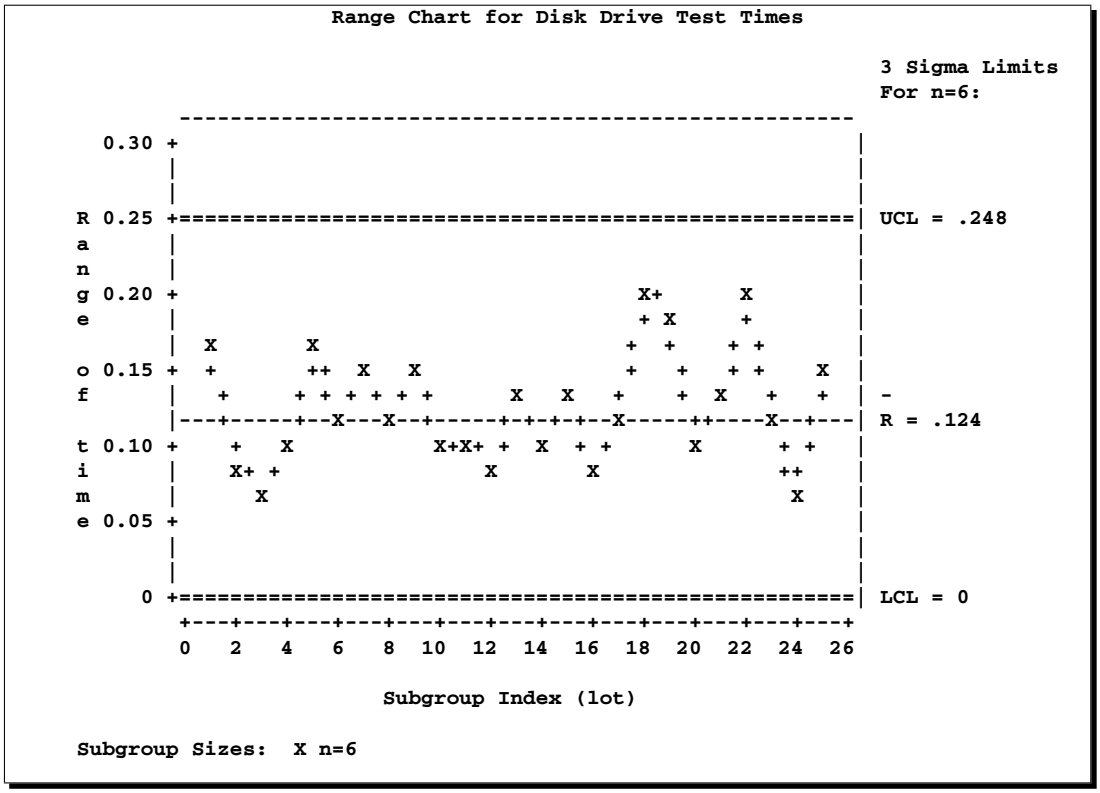


Figure 46.4. R Chart from the Summary Data Set DISKSUM

In general, a HISTORY= input data set used with the RCHART statement must contain the following variables:

- subgroup variable
- subgroup range variable
- subgroup sample size variable

Furthermore, the names of the subgroup range and sample size variables must begin with the *process* name specified in the RCHART statement and end with the special suffix characters *R* and *N*, respectively. If the names do not follow this convention, you can use the RENAME option in the PROC SHEWHART statement to rename the variables for the duration of the SHEWHART procedure step (see page 1743).

In summary, the interpretation of *process* depends on the input data set.

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names of the variables containing the summary statistics.

For more information, see "HISTORY= Data Set" on page 1601.

Saving Summary Statistics

In this example, the RCHART statement procedure is used to create a summary data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read measurements from the data set DISKS and create a summary data set named DISKHIST:

See SHWRCHR in the SAS/QC Sample Library

```
proc shewhart data=disks;
   rchart time*lot / outhistory = diskhist
                     nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 46.2. Options such as OUTHISTORY= and NOCHART are specified after the slash (/) in the RCHART statement. A complete list of options is presented in the "Syntax" section on page 1585.

Figure 46.5 contains a partial listing of DISKHIST.

Summary Data Set for Disk Times

lot	timeX	time R	time N
1	8.00833	0.16	6
2	8.02167	0.09	6
3	7.97833	0.07	6
4	8.00667	0.10	6
5	8.01833	0.16	6
.	.	.	.
.	.	.	.
.	.	.	.

Figure 46.5. The Summary Data Set DISKHIST

There are four variables in the data set DISKHIST.

- LOT contains the subgroup index.
- TIMEX contains the subgroup means.
- TIMER contains the subgroup ranges.
- TIMEN contains the subgroup sample sizes.

The subgroup mean variable is included in the OUTHISTORY= data set even though it is not required by the RCHART statement. This allows the data set to be used as a HISTORY= data set with the BOXCHART, XCHART, and XRCHART statements, as well as with the RCHART statement. Note that the summary statistic variables are named by adding the suffix characters *X*, *R*, and *N* to the *process* TIME specified in the RCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 1598.

Saving Control Limits

See SHWRCHR in the SAS/QC Sample Library

You can save the control limits for an R chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1583) or modify the limits with a DATA step program.

The following statements read measurements from the data set DISKS (see page 1574) and save the control limits displayed in Figure 46.2 in a data set named DISKLIM:

```
title 'Control Limits for Disk Times';
proc shewhart data=disks;
   rchart time*lot / outlimits = disklim
                     nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. The data set DISKLIM is listed in Figure 46.6.

Control Limits for Disk Times

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_
time	lot	ESTIMATE	6	.004447667	3

LCLX	_MEAN_	_UCLX_	_LCLR_	_R_	_UCLR_	_STDDEV_
7.94314	8.00307	8.06299	0	0.124	0.24847	0.048927

Figure 46.6. The Data Set DISKLIM Containing Control Limit Information

The data set DISKLIM contains one observation with the limits for *process* TIME. The variables _LCLR_ and _UCLR_ contain the lower and upper control limits, and the variable _R_ contains the central line. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values. The variables _LCLX_ and _UCLX_, which contain the lower and upper control limits for subgroup means, are included so that the data set DISKLIM can be used to create an $\bar{X}$ chart (see Chapter 50, "XRCHART Statement,"). For more information, see "OUTLIMITS= Data Set" on page 1596.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
title 'Summary Statistics and Control Limit Information';
proc shewhart data=disks;
  rchart time*lot / outtable=disktab
                    nochart;
run;
```

The data set DISKTAB is listed in Figure 46.7.

Summary Statistics and Control Limit Information

VAR	lot	_SIGMAS_	_LIMITN_	_SUBN_	_LCLR_	_SUBR_	_R_	_UCLR_	_EXLIM_
time	1	3	6	6	0	0.16	0.124	0.24847	
time	2	3	6	6	0	0.09	0.124	0.24847	
time	3	3	6	6	0	0.07	0.124	0.24847	
time	4	3	6	6	0	0.10	0.124	0.24847	
time	5	3	6	6	0	0.16	0.124	0.24847	
time	6	3	6	6	0	0.12	0.124	0.24847	
time	7	3	6	6	0	0.15	0.124	0.24847	
time	8	3	6	6	0	0.11	0.124	0.24847	
time	9	3	6	6	0	0.15	0.124	0.24847	
time	10	3	6	6	0	0.10	0.124	0.24847	
time	11	3	6	6	0	0.10	0.124	0.24847	
time	12	3	6	6	0	0.09	0.124	0.24847	
time	13	3	6	6	0	0.13	0.124	0.24847	
time	14	3	6	6	0	0.10	0.124	0.24847	
time	15	3	6	6	0	0.13	0.124	0.24847	
time	16	3	6	6	0	0.09	0.124	0.24847	
time	17	3	6	6	0	0.11	0.124	0.24847	
time	18	3	6	6	0	0.20	0.124	0.24847	
time	19	3	6	6	0	0.18	0.124	0.24847	
time	20	3	6	6	0	0.10	0.124	0.24847	
time	21	3	6	6	0	0.14	0.124	0.24847	
time	22	3	6	6	0	0.20	0.124	0.24847	
time	23	3	6	6	0	0.11	0.124	0.24847	
time	24	3	6	6	0	0.06	0.124	0.24847	
time	25	3	6	6	0	0.15	0.124	0.24847	

Figure 46.7. The Data Set DISKTAB

This data set contains one observation for each subgroup sample. The variables _SUBR_ and _SUBN_ contain the subgroup ranges and subgroup sample sizes. The variables _LCLR_ and _UCLR_ contain the lower and upper control limits, and the variable _R_ contains the central line. The variables _VAR_ and BATCH contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1599. An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read DISKTAB and display an R chart (not shown here) identical to the chart in Figure 46.2:

```
title 'Range Chart for Disk Drive Test Times';
proc shewhart table=disktab;
   rchart time*lot;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see Chapter 56, "Specialized Control Charts,"). For more information, see "TABLE= Data Set" on page 1602.

Reading Preestablished Control Limits

See SHWRCHR in the SAS/QC Sample Library

In the previous example, the OUTLIMITS= data set DISKLIM saved control limits computed from the measurements in DISKS. This example shows how these limits can be applied to new data provided in the following data set:

```
data disks2;
  input lot @;
  do i=1 to 6;
     input time @;
     output;
     end;
  drop i;
datalines;
26 7.93 7.97 7.89 7.81 7.88 7.92
27 7.86 7.91 7.87 7.89 7.83 7.87
28 7.93 7.95 7.90 7.89 7.88 7.90
29 7.97 8.00 7.86 7.89 7.84 7.78
30 7.91 7.93 7.98 7.93 7.83 7.88
31 7.85 7.94 7.88 7.98 7.96 7.84
32 7.86 8.01 7.88 7.95 7.90 7.89
33 7.87 7.93 7.96 7.89 7.81 8.00
34 7.87 7.97 7.95 7.89 7.92 7.84
35 7.92 7.97 7.90 7.88 7.89 7.86
36 7.96 7.90 7.90 7.84 7.90 8.00
37 7.92 7.90 7.98 7.92 7.94 7.94
38 7.88 7.99 8.02 7.98 7.88 7.92
39 7.89 7.91 7.92 7.90 7.94 7.94
40 7.84 7.88 7.91 7.98 7.87 7.93
41 7.91 7.87 7.96 7.91 7.89 7.92
42 7.96 7.93 7.86 7.93 7.86 7.94
43 7.84 7.82 7.87 7.91 7.91 8.01
44 7.93 7.91 7.92 7.88 7.91 7.86
45 7.95 7.92 7.93 7.90 7.86 8.00
;
```

The following statements create an R chart using the control limits in DISKLIM:

```
title 'Range Chart for Disk Drive Test Times';
proc shewhart data=disks2 limits=disklim;
   rchart time*lot;
run;
```

The chart is shown in Figure 46.8. The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default,* this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name TIME
- the value of _SUBGRP_ matches the *subgroup-variable* name LOT

*In Release 6.09 and in earlier releases, it is also necessary to specify the READLIMITS option to read control limits from a LIMITS= data set.

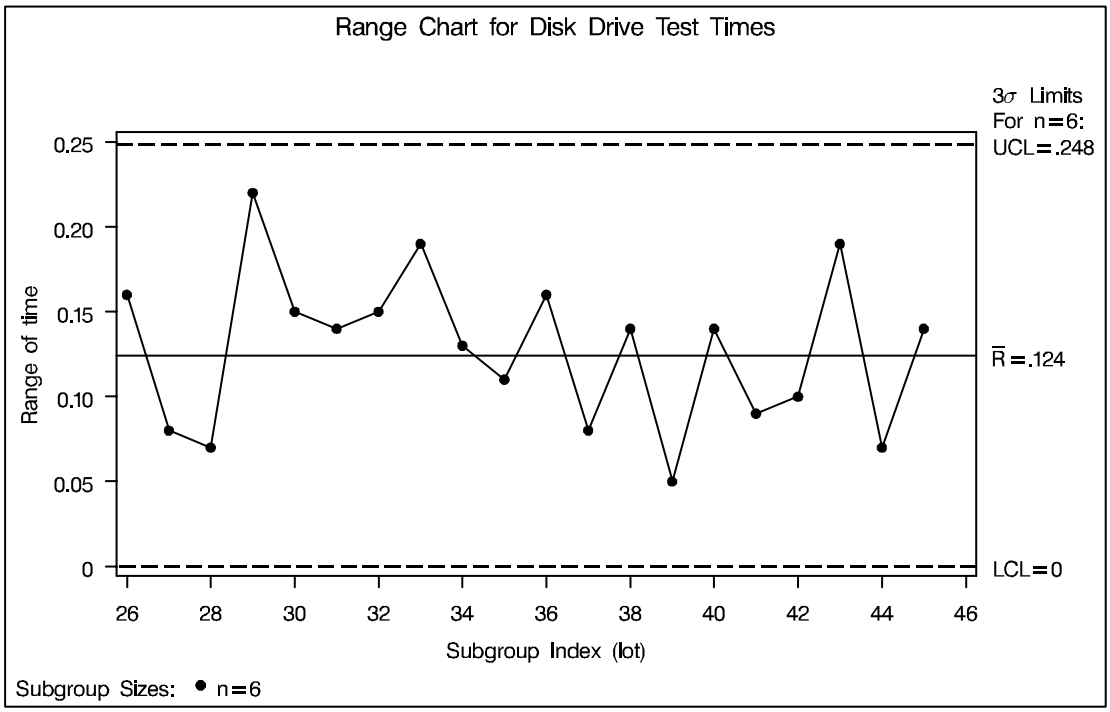


Figure 46.8. R Chart for Second Set of Disk Drive Test Times

All the ranges lie within the control limits, indicating that the variability in disk drive performance is still in statistical control.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See Example 46.2 on page 1607 and "LIMITS= Data Set" on page 1601 for details concerning the variables that you must provide.

Syntax

The basic syntax for the RCHART statement is as follows:

RCHART *process******subgroup-variable* **;**

The general form of this syntax is as follows:

RCHART (*processes***)***subgroup-variable* <**(***block-variables* **)** >
< =*symbol-variable* | =*'character'* > < **/** *options* >**;**

You can use any number of RCHART statements in the SHEWHART procedure. The components of the RCHART statement are described as follows.

process
processes
identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating Range Charts from Raw Data" on page 1574.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating Range Charts from Summary Data" on page 1576.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1580.

A *process* is required. If you specify more than one *process*, enclose the list in parentheses. For example, the following statements request distinct R charts for WEIGHT, LENGTH, and WIDTH:

```
proc shewhart data=measures;
   rchart (weight length width)*day;
run;
```

subgroup-variable
is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding RCHART statement, DAY is the subgroup variable. For details, see "Subgroup Variables" on page 1771.

block-variables
are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 1932 for an example.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the ranges.

- If you produce a chart on a line printer, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce a chart on a graphics device, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOLn statements. See "Displaying Stratification in Levels of a Classification Variable" on page 1931 for an example.

character
specifies a plotting character for charts produced on line printers. For example, the following statements create an R chart using an asterisk (*) to plot the points:

```
proc shewhart data=values;
   rchart weight*day='*';
run;
```

options
enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The "Summary of Options" section, which follows, lists all options by function. Chapter 53, "Dictionary of Options," describes each option in detail.

Summary of Options

The following tables list the RCHART statement options by function. For complete descriptions, see Chapter 53, "Dictionary of Options."

Table 46.1. Tabulation Options

TABLE	creates a basic table of subgroup values, subgroup sample sizes, subgroup ranges, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	augments basic table with value of central line
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with a column indicating which tests for special causes are positive

Note that specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.

Table 46.2. Options for Specifying Tests for Special Causes

NO3SIGMACHECK	allows tests to be applied with control limits other than 3σ limits
TESTS2=*value-list*| *customized-pattern-list*	specifies tests for special causes for the R chart
TEST2RESET=*variable*	allows tests for special causes to be reset for the R chart
TEST2RUN=*n*	specifies length of pattern for Test 2
TEST3RUN=*n*	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL='*label*'| (*variable*)|*keyword*	provides labels for points where test is positive
TESTLABEL*n*='*label*'	specifies label for n^{th} test for special causes
TESTNMETHOD= STANDARDIZE	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
ZONE2LABELS	adds labels A, B, and C to zone lines
ZONE2VALUES	labels zone lines with their values
ZONES2	adds lines delineating zones A, B, and C
ZONEVALPOS=*n*	specifies position of ZONE2VALUES labels

Table 46.3. Graphical Options for Displaying Tests for Special Causes

CTESTLABBOX=*color*	specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=*color*| *test-color-list*	specifies color for labels indicating points where test is positive
CTESTSYMBOL=*color*	specifies color for symbol used to plot points where test is positive
CZONES=*color*	specifies color for lines and labels delineating zones A, B, and C
LTESTS=*line-type*	specifies type of line connecting points where test is positive
LZONES=*linetype*	specifies line type for lines delineating zones A, B, and C
TESTFONT=*font*	specifies software font for labels at points where test is positive
TESTHEIGHT=*value*	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=*symbol*	specifies plot symbol for points where test is positive
TESTSYMBOLHT=*value*	specifies symbol height for points where test is positive
WTESTS=*n*	specifies width of line connecting points where test is positive

Table 46.4. Line Printer Options for Displaying Tests for Special Causes

TESTCHAR='*character*'	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
ZONECHAR='*character*'	specifies character for lines that delineate zones for tests for special causes

Table 46.5. Clipping Options

CCLIP=*color*	specifies color for plot symbol for clipped points
CLIPCHAR='*character*'	specifies plot character for clipped points
CLIPFACTOR=*value*	determines extent to which extreme points are clipped
CLIPLEGEND='*string*'	specifies text for clipping legend
CLIPLEGPOS=*keyword*	specifies position of clipping legend
CLIPSUBCHAR='*character*'	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=*symbol*	specifies plot symbol for clipped points
CLIPSYMBOLHT=*value*	specifies symbol marker height for clipped points

Table 46.6. Reference Line Options

CHREF=*color*	specifies color for lines requested by the HREF= option
CVREF=*color*	specifies color for lines requested by the VREF= option
HREF=*values*|*SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis
HREFCHAR='*character*'	specifies line character for HREF= lines
HREFDATA=*SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on individual measurements chart
HREFLABELS=*'label1'...'labeln'*	specifies labels for HREF= lines
HREFLABPOS=*n*	specifies position of HREFLABELS= labels
LHREF=*line-type*	specifies line type for HREF= lines
LVREF=*line-type*	specifies line type for VREF= lines
NOBYREF	specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=*values*|*SAS-data-set*	specifies position of reference lines perpendicular to vertical axis
VREFCHAR='*character*'	specifies line character for VREF= lines
VREFLABELS=*'label1'...'labeln'*	specifies labels for VREF= lines
VREFLABPOS=*n*	specifies position of VREFLABELS= labels

Table 46.7. Axis and Axis Label Options

CAXIS=*color*	specifies color for axis lines and tick marks
CFRAME=*color*|(*color-list*)	specifies fill colors for frame for plot area
CTEXT=*color*	specifies color for tick mark values and axis labels
HAXIS=*values*|AXIS*n*	specifies major tick mark values for horizontal axis
HEIGHT=*value*	specifies height of axis label and axis legend text
HMINOR=*n*	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=*value*	specifies length of offset at both ends of horizontal axis
INTSTART=*value*	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	suppresses vertical axis truncation at zero applied by default
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=*n*	specifies thinning factor for tick mark labels on horizontal axis
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=*values*|AXIS*n*	specifies major tick mark values for vertical axis
VFORMAT=*format*	specifies format for vertical axis tick mark labels
VMINOR=*n*	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=*value*	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis
WAXIS=*n*	specifies width of axis lines

Table 46.8. Options for Specifying Control Limits

ALPHA=*value*	requests probability limits for control charts
LIMITN=*n*|VARYING	specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	computes control limits for each *process* from the data rather than from a LIMITS= data set (Release 6.10 and later releases)
READALPHA	reads the variable _ALPHA_ instead of the variable _SIGMAS_ from a LIMITS= data set
READINDEXES=ALL|'*label1*'...'*labeln*'	reads multiple sets of control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (Release 6.09 and earlier releases)
SIGMAS=*k*	specifies width of control limits in terms of multiple k of standard error of plotted ranges

Table 46.9. Options for Displaying Control Limits

CINFILL=*color*	specifies color for area inside control limits
CLIMITS=*color*	specifies color of control limits, central line, and related labels
LCLLABEL='*label*'	specifies label for lower control limit
LIMLABSUBCHAR= '*character*'	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=*linetype*	specifies line type for control limits
NDECIMAL=*n*	specifies number of digits to right of decimal place in default labels for control limits and central line
NOCTL	suppresses display of central line
NOLCL	suppresses display of lower control limit
NOLIMITLABEL	suppresses labels for control limits and central line
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOLIMIT0	suppresses display of zero lower control limit
NOUCL	suppresses display of upper control limit
RSYMBOL='*string*' | *keyword*	specifies label for central line
UCLLABEL='*string*'	specifies label for upper control limit
WLIMITS=*n*	specifies width for control limits and central line

Table 46.10. Process Mean and Standard Deviation Options

SIGMA0=*value*	specifies known value σ_0 for process standard deviation σ
SMETHOD=*keyword*	specifies method for estimating process standard deviation σ
TYPE=*keyword*	identifies whether parameters are estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set

Table 46.11. Specification Limit Options

CIINDICES (ALPHA=*value* TYPE=*keyword*)	specifies α value and type for computing capability index confidence limits
LSL=*value-list*	specifies list of lower specification limits
TARGET=*value-list*	specifies list of target values
USL=*value-list*	specifies list of upper specification limits

Table 46.12. Block Variable Legend Options

BLOCKLABELPOS=*keyword*	specifies position of label for *block-variable* legend
BLOCKLABTYPE=*value*|*keyword*	specifies text size of *block-variable* legend
BLOCKPOS=*n*	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=*color-list*	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=*variable*|(*variables*)	specifies one or more variables whose values are colors for filling background of *block-variable* legend

Table 46.13. Grid Options

CGRID=*color*	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=*linetype*	specifies line type for grid requested with the ENDGRID option
LGRID=*linetype*	specifies line type for grid requested with the GRID option
WGRID=*n*	specifies width of grid lines requested

Table 46.14. Options for Plotting and Labeling Points

ALLLABEL2=VALUE|(*variable*)	labels every point on chart
CLABEL=*color*	specifies color for labels
CCONNECT=*color*	specifies color for line segments that connect points on chart
CFRAMELAB=*color*	specifies fill color for frame around labeled points
CNEEDLES=*color*	specifies color for needles that connect points to central line
CONNECTCHAR='*character*'	specifies character used to form line segments that connect points on chart
COUT=*color*	specifies color for portions of line segments that connect points outside control limits
COUTFILL=*color*	specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=*angle*	specifies angle at which labels are drawn
LABELFONT=*font*	specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=*value*	specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on chart
OUTLABEL2=VALUE|(*variable*)	labels points outside control limits
SYMBOLCHARS='*characters*'	specifies characters indicating *symbol-variable*
SYMBOLLEGEND=NONE|*name*	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=*keyword*	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=*n*	specifies width of needles

Table 46.15. Phase Options

CPHASELEG=*color*	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE='*string*'	specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=*value*|*keyword*	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES= ALL|'*label1*'...'*labeln*'	specifies *phases* to be read from an input data set

Table 46.16. Input Data Set Options

MISSBREAK	specifies that observations with missing values are not to be processed

Table 46.17. Output Data Set Options

OUTHISTORY= *SAS-data-set*	creates output data set containing subgroup summary statistics
OUTINDEX='*string*'	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS= *SAS-data-set*	creates output data set containing control limits
OUTTABLE= *SAS-data-set*	creates output data set containing subgroup summary statistics and control limits

Table 46.18. Plot Layout Options

ALLN	plots ranges for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts for a process variable only when exceptions occur
INTERVAL=*keyword*	specifies natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=*n*	specifies maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=*n*	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=*n*	specifies number of pages or screens to be used to display chart
ZEROSTD	displays R chart regardless of whether $\hat{\sigma} = 0$

Table 46.19. Graphical Enhancement Options

ANNOTATE=*SAS-data-set*	specifies annotate data set that adds features to chart
DESCRIPTION='*string*'	specifies string that appears in the description field of the PROC GREPLAY master menu
FONT=*font*	specifies software font for labels and legends on chart
NAME='*string*'	specifies name that appears in the name field of the PROC GREPLAY master menu
PAGENUM='*string*'	specifies the form of the label used in pagination
PAGENUMPOS= *keyword*	specifies the position of the page number requested with the PAGENUM= option

Table 46.20. Star Options

CSTARCIRCLES=*color*	specifies color for STARCIRCLES= circles
CSTARFILL=*color*| *(variable)*	specifies color for filling stars
CSTAROUT=*color*	specifies outline color for stars exceeding inner or outer circles
CSTARS=*color*| *(variable)*	specifies color for outlines of stars
LSTARCIRCLES= *linetypes*	specifies line types for STARCIRCLES= circles
LSTARS=*linetype*|*(variable)*	specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=*value*	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=*value-list*	specifies reference circles for stars
STARINRADIUS=*value*	specifies inner radius of stars
STARLABEL=*keyword*	specifies vertices to be labeled
STARLEGEND=*keyword*	specifies style of legend for star vertices
STARLEGENDLAB=*'label'*	specifies label for STARLEGEND= legend
STAROUTRADIUS=*value*	specifies outer radius of stars
STARSPEC=*value*| *SAS-data-set*	specifies method used to standardize vertex variables
STARSTART=*value*	specifies angle for first vertex
STARTYPE=*keyword*	specifies graphical style of star
STARVERTICES= *variable*|*(variables)*	superimposes star at each point on chart
WSTARCIRCLES=*n*	specifies width of STARCIRCLES= circles
WSTARS=*n*	specifies width of STARVERTICES= stars

Table 46.21. Overlay Options

CCOVERLAY=*color-list*	specifies colors for overlay line segments
COVERLAY=*color-list*	specifies colors for overlay plots
COVERLAYCLIP=*color*	specifies color for clipped points on overlays
LOVERLAY=*linetypes*	specifies line types for overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=*variable-list*	specifies variables to overlay on control chart
OVERLAYCLIPSYM=*symbol*	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=*value*	specifies symbol height for clipped points on overlays
OVERLAYHTML=*variable-list*	specifies URLS to associate with overlay points
OVERLAYID=*variable-list*	specifies labels for overlay points
OVERLAYLEGLAB='*label*'	specifies label for overlay legend
OVERLAYSYM=*symbol-list*	specifies symbols for overlay plots
OVERLAYSYMHT=*value-list*	specifies symbol heights for overlay plots
WOVERLAY=*value-list*	specifies widths of overlay line segments

Table 46.22. Options for Interactive Control Charts

HTML=(*variable*)	specifies a variable whose values are URLs to be associated with subgroups
HTML_LEGEND=(*variable*)	specifies a variable whose values are URLs to be associated with symbols in the symbol legend
WEBOUT=*SAS-data-set*	creates an OUTTABLE= data set with additional graphics coordinate data

Details

Constructing Range Charts

The following notation is used in this section:

σ	process standard deviation (standard deviation of the population of measurements)
R_i	range of measurements in i^{th} subgroup
n_i	sample size of i^{th} subgroup
$d_2(n)$	expected value of the range of n independent normally distributed variables with unit standard deviation
$d_3(n)$	standard error of the range of n independent observations from a normal population with unit standard deviation
$D_p(n)$	$100p^{\text{th}}$ percentile of the distribution of the range of n independent observations from a normal population with unit standard deviation

Plotted Points

Each point on an R chart indicates the value of a subgroup range (R_i). For example, if the tenth subgroup contains the values 12, 15, 19, 16, and 14, the value plotted for this subgroup is $R_{10} = 19 - 12 = 7$.

Central Line

By default, the central line for the i^{th} subgroup indicates an estimate of the expected value of R_i, which is computed as $d_2(n_i)\hat{\sigma}$, where $\hat{\sigma}$ is an estimate of σ. If you specify a known value (σ_0) for σ, the central line indicates the value of $d_2(n_i)\sigma_0$. Note that the central line varies with n_i.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of R_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that R_i exceeds the limits

The following table provides the formulas for the limits:

Table 46.23. Limits for R Charts

Control Limits
LCL = lower limit = $\max(d_2(n_i)\hat{\sigma} - kd_3(n_i)\hat{\sigma}, 0)$
UCL = upper limit = $d_2(n_i)\hat{\sigma} + kd_3(n_i)\hat{\sigma}$

Probability Limits
LCL = lower limit = $D_{\alpha/2}\hat{\sigma}$
UCL = upper limit = $D_{1-\alpha/2}\hat{\sigma}$

The formulas assume that the data are normally distributed. Note that the control limits vary with n_i and that the probability limits for R_i are asymmetric around the central line. If a standard value σ_0 is available for σ, replace $\hat{\sigma}$ with σ_0 in Table 46.23.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables are saved:

Table 46.24. OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
_CP_	capability index C_p
CPK	capability index C_{pk}
CPL	capability index CPL
CPM	capability index C_{pm}
CPU	capability index CPU
INDEX	optional identifier for the control limits with the OUTINDEX= option
LCLR	lower control limit for subgroup range
LCLX	lower control limit for subgroup mean
LIMITN	sample size associated with the control limits
LSL	lower specification limit
MEAN	process mean ($\overline{\overline{X}}$)
_R_	value of central line on R chart
SIGMAS	multiple (k) of standard error of R_i
STDDEV	process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*subgroup-variable* specified in the RCHART statement
TARGET	target value
TYPE	type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLR	upper control limit for subgroup range
UCLX	upper control limit for subgroup mean
USL	upper specification limit
VAR	*process* specified in the RCHART statement

Notes:

1. The variables _LCLX_, _MEAN_, and _UCLX_ are saved to allow the OUTLIMITS= data set to be used as a LIMITS= data set with the BOXCHART, XCHART, and XRCHART statements.
2. If the control limits vary with subgroup sample size, the special missing value *V* is assigned to the variables _LIMITN_, _LCLX_, _UCLX_, _LCLR_, _R_, and _UCLR_.
3. If the limits are defined in terms of a multiple k of the standard error of R_i, the value of _ALPHA_ is computed as

$$F_R(\text{\_LCLR\_}/\text{\_STDDEV\_}) + 1 - F_R(\text{\_UCLR\_}/\text{\_STDDEV\_})$$

   where $F_R(\cdot)$ is the cumulative distribution function of the range of a sample of n observations from a normal population with unit standard deviation, and n is the value of _LIMITN_. If _LIMITN_ has the special missing value *V*, this value is assigned to _ALPHA_.
4. If the limits are probability limits, the value of _SIGMAS_ is computed as $(\text{\_UCLR\_} - \text{\_R\_})/e$, where e is the standard error of the range of n observations from a normal population with unit standard deviation. If _LIMITN_ has the special missing value *V*, this value is assigned to _SIGMAS_.

5. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you provide a target value with the TARGET= option. See "Capability Indices" on page 1774 for computational details.
6. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the RCHART statement. For an example, see "Saving Control Limits" on page 1580.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables are saved:

- the *subgroup-variable*
- a subgroup mean variable named by *process* suffixed with *X*
- a subgroup range variable named by *process* suffixed with *R*
- a subgroup sample size variable named by *process* suffixed with *N*

The subgroup mean variable is saved so that the data set can be reused as a HISTORY= data set with the BOXCHART, XCHART, and XRCHART statements, as well as the RCHART statement.

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the RCHART statement. For example, consider the following statements:

```
proc shewhart data=steel;
   rchart (width diameter)*lot / outhistory=summary;
run;
```

The data set SUMMARY contains variables named LOT, WIDTHX, WIDTHR, WIDTHN, DIAMTERX, DIAMTERR, and DIAMTERN. Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 1579.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables are saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on R chart
LCLR	lower control limit for range
LIMITN	nominal sample size associated with the control limits
_R_	average range
SIGMAS	multiple (k) of the standard error associated with control limits
subgroup	values of the subgroup variable
SUBN	subgroup sample sizes
SUBR	subgroup range
TESTS2	tests for special causes signaled on R chart
UCLR	upper control limit for range
VAR	*process* specified in the RCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved, depending on how the control limits are defined (with the ALPHA= or SIGMAS= option, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS2_ is saved if you specify the TESTS2= option.
3. The variables _EXLIM_ and _TESTS2_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1580.

ODS Tables

The following table summarizes the ODS tables that you can request with the RCHART statement.

Table 46.25. ODS Tables Produced with the RCHART Statement

Table Name	Description	Options
RCHART	R chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the RCHART statement must be a SAS variable in the DATA= data set. This variable provides measurements that must be grouped in subgroup samples indexed by the *subgroup-variable*. The *subgroup-variable*, which is specified in the RCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a raw measurement for each *process* and a value for the *subgroup-variable*. If the i^{th} subgroup contains n_i items, there should be n_i consecutive observations for which the value of the subgroup variable is the index of the i^{th} subgroup. For example, if each subgroup contains five items and there are 30 subgroup samples, the DATA= data set should contain 150 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the DATA= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 1936).

For an example of a DATA= data set, see "Creating Range Charts from Raw Data" on page 1574.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set CONLIMS:*

```
proc shewhart data=info limits=conlims;
   rchart weight*batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set; see Table 46.24 on page 1596. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLR_, _R_, and _UCLR_, which specify the control limits directly
- the variable _STDDEV_, which is used to calculate the control limits according to the equations in Table 46.23 on page 1596

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are `ESTIMATE`, `STANDARD`, `STDMU`, and `STDSIGMA`.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1583.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This allows you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART procedure or to read output data sets created with SAS summarization procedures, such as PROC MEANS.

A HISTORY= data set used with the RCHART statement must contain the following:

- the *subgroup-variable*
- a subgroup range variable for each *process*
- a subgroup sample size variable for each *process*

*In Release 6.09 and in earlier releases, it is necessary to specify the READLIMITS option.

The names of the subgroup range and subgroup sample size variables must be the prefix *process* concatenated with the special suffix characters R and N, respectively.

For example, consider the following statements:

```
proc shewhart history=summary;
   rchart (weight yldstren)*batch;
run;
```

The data set SUMMARY must include the variables BATCH, WEIGHTR, WEIGHTN, YLDSRENR, and YLDSRENN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as phases) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 1936 for an example).

For an example of a HISTORY= data set, see "Creating Range Charts from Summary Data" on page 1576.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in Chapter 56, "Specialized Control Charts."

The following table lists the variables required in a TABLE= data set used with the RCHART statement:

Table 46.26. Variables Required in a TABLE= Data Set

Variable	Description
LCLR	lower control limit for range
LIMITN	nominal sample size associated with the control limits
_R_	average range
subgroup-variable	values of the *subgroup-variable*
SUBN	subgroup sample size
SUBR	subgroup range
UCLR	upper control limit for range

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS2_ (if the TESTS2= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1580.

Methods for Estimating the Standard Deviation

When control limits are determined from the input data, two methods (referred to as default and MVLUE) are available for estimating σ.

Default Method

The default estimate for σ is

$$\hat{\sigma} = \frac{R_1/d_2(n_1) + \cdots + R_N/d_2(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, and R_i is the sample range of the observations $x_{i1}, \ldots, x_{in_i}$ in the i^{th} subgroup.

$$R_i = \max_{1 \leq j \leq n_i}(x_{ij}) - \min_{1 \leq j \leq n_i}(x_{ij})$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$. The unbiasing factor $d_2(n_i)$ is defined so that, if the observations are normally distributed, the expected value of R_i is $d_2(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the *ASTM Manual on Presentation of Data and Control Chart Analysis* (1976).

MVLUE Method

If you specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). The MVLUE is a weighted average of N unbiased estimates of σ of the form $R_i/d_2(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{f_1 R_1/d_2(n_1) + \cdots + f_N R_N/d_2(n_N)}{f_1 + \cdots + f_N}$$

where

$$f_i = \frac{[d_2(n_i)]^2}{[d_3(n_i)]^2}$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The unbiasing factor $d_3(n_i)$ is defined so that, if the observations are normally distributed, the expected value of σ_{R_i} is $d_3(n_i)\sigma$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical	DATA=	*process*
Vertical	HISTORY=	subgroup range variable
Vertical	TABLE=	_SUBR_

For an example, see "Labeling Axes" on page 1966.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples

This section provides advanced examples of the RCHART statement.

Example 46.1. Computing Probability Limits

See SHWREX1 in the SAS/QC Sample Library

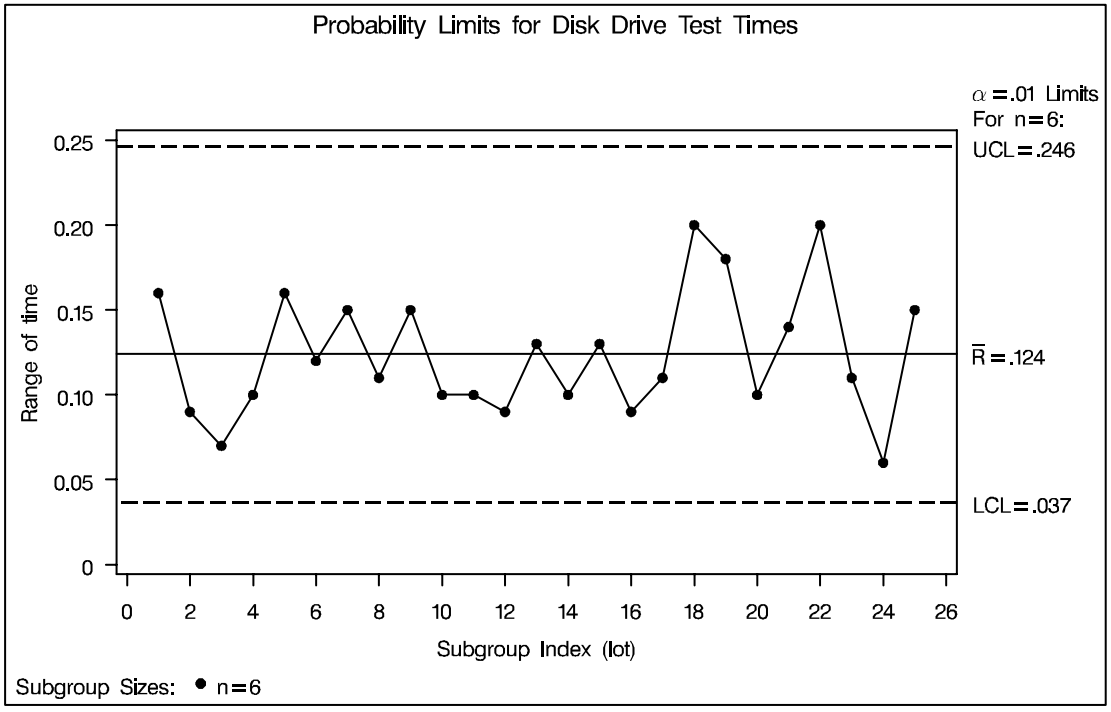
This example demonstrates how to create R charts with probability limits. The following statements read the disk drive test times from the data set DISKS (see page 1574) and create the R chart shown in Output 46.1.1:

```
title 'Probability Limits for Disk Drive Test Times';
proc shewhart data=disks;
   rchart time*lot / alpha     = .01
                     outlimits = dlimits;
run;
```

The ALPHA= option specifies the probability (α) that a subgroup range exceeds its limits. Here, the limits are computed so that the probability that a range is less than the lower limit is $\alpha/2 = 0.005$, and the probability that a range is greater than the upper limit is $\alpha/2 = 0.005$. This assumes that the measurements are normally distributed. The OUTLIMITS= option names an output data set that saves the probability limits. A listing of DLIMITS is shown in Output 46.1.2.

The variable _ALPHA_ saves the value of α. Note that, in this case, the upper probability limit is equivalent to an upper 2.95σ limit.

Output 46.1.1. R Chart with Probability Limits



Output 46.1.2. Probability Limits Data Set

Probablity Limits for Disk Drive Test Times

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_
time	lot	ESTIMATE	6	0.01	2.94715	7.95162

MEAN	_UCLX_	_LCLR_	_R_	_UCLR_	_STDDEV_
8.00307	8.05452	0.036645	0.124	0.24628	0.048927

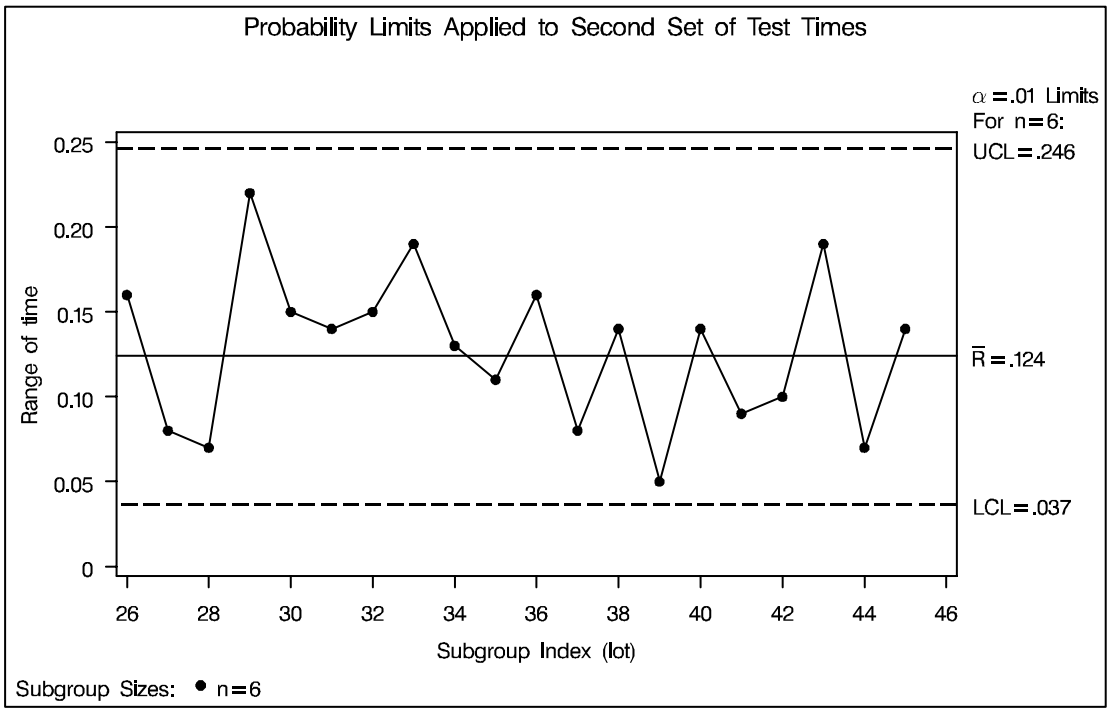
Since all the points fall within the probability limits, it can be concluded that the variability in the disk drive performance is in statistical control.

The following statements apply the limits in DLIMITS to the times in the data set DISKS2 (see page 1583):

```
title 'Probability Limits Applied to Second Set of Test Times';
proc shewhart data=disks2 limits=dlimits;
   rchart time*lot / readalpha;
run;
```

The READALPHA option* specifies that the variable _ALPHA_, rather than the variable _SIGMAS_, is to be read from the LIMITS= data set. Thus the limits displayed in the chart, shown in Output 46.1.3, are probability limits.

Output 46.1.3. Reading Probability Limits from a LIMITS= Data Set



*In Release 6.09 and in earlier releases, it is also necessary to specify the READLIMITS option.

Example 46.2. Specifying Control Limit Information

This example illustrates how you can use a DATA step program to create a LIMITS= data set. You can provide previously established values for the limits and central line with the variables _LCLR_, _R_, and _UCLR_, as in the following statements:

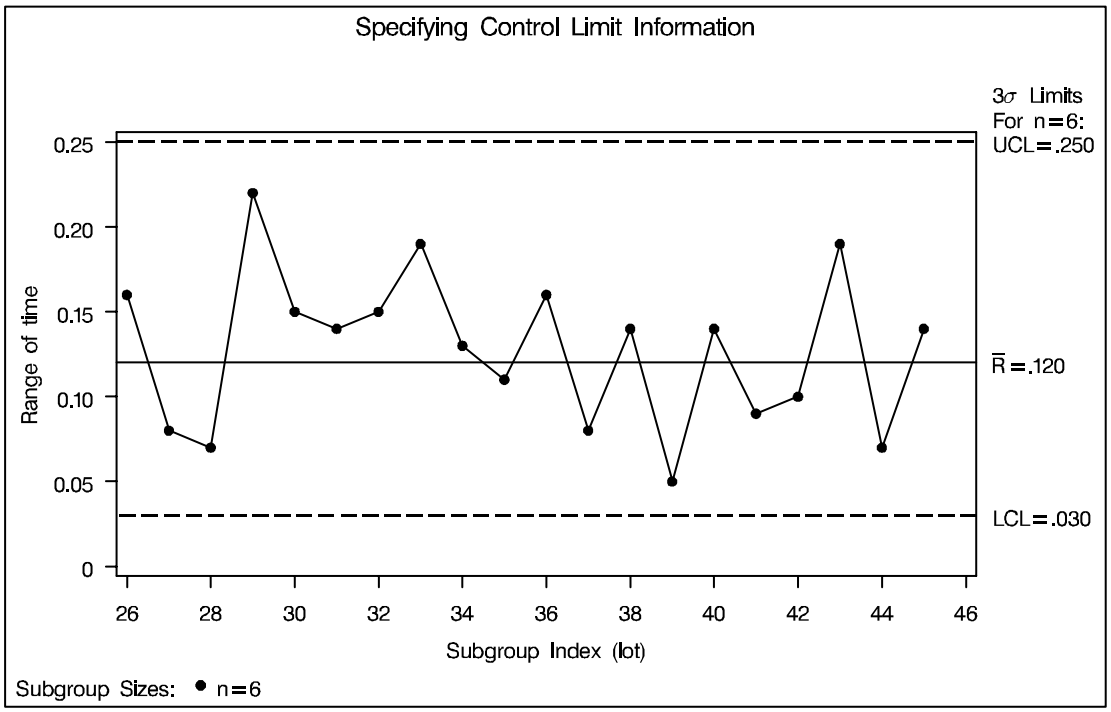
See SHWREX2 in the SAS/QC Sample Library

```
data dlimits2;
   length _var_ _subgrp_ _type_ $8;
   _var_    = 'time';
   _subgrp_ = 'lot';
   _type_   = 'STANDARD';
   _limitn_ = 6;
   _lclr_   = .03;
   _r_      = .12;
   _uclr_   = .25;
run;
```

The following statements* apply the control limits in DLIMITS2 to the measurements in DISKS2 (see page 1583) and create the R chart shown in Output 46.2.1:

```
title 'Specifying Control Limit Information';
proc shewhart data=disks2 limits=dlimits2;
   rchart time*lot;
run;
```

Output 46.2.1. Reading Control Limits from DLIMITS2



*In Release 6.09 and in earlier releases, it is also necessary to specify the READLIMITS option.

In some cases, a standard value (σ_0) may be available for the process standard deviation. The following DATA step creates a data set named DLIMITS3 that provides this value:

```
data dlimits3;
   length _var_ _subgrp_ _type_ $8;
   _var_    = 'time';
   _subgrp_ = 'lot';
   _stddev_ = .045;
   _limitn_ = 6;
   _type_   = 'STDSIGMA';
run;
```

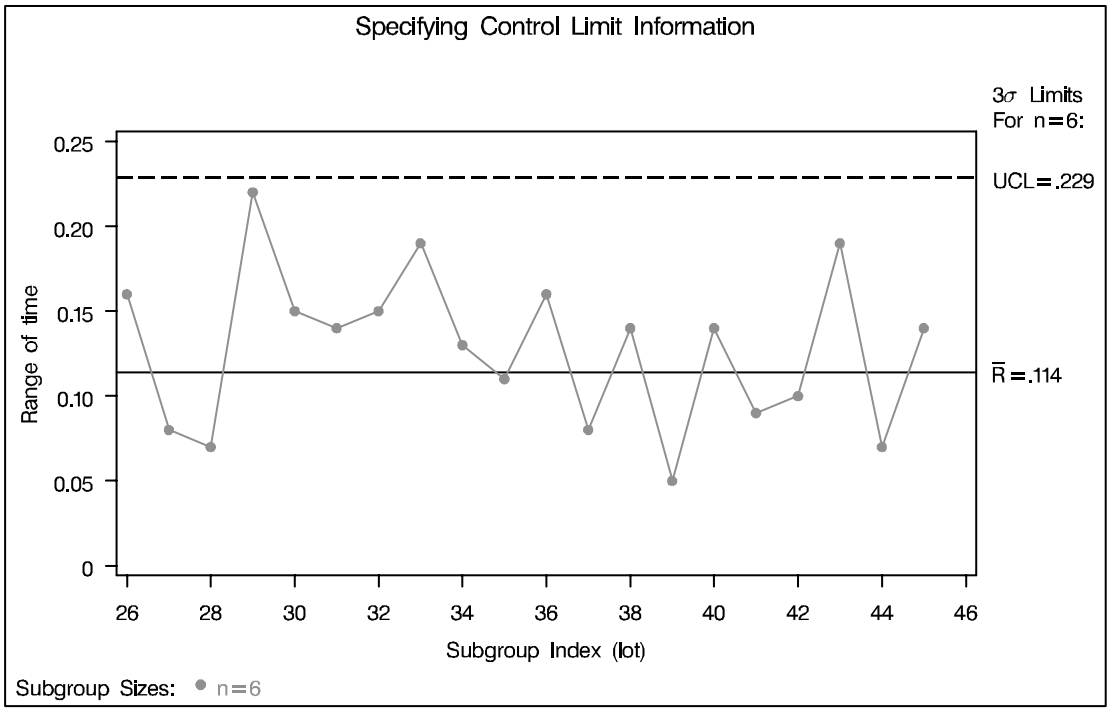
The variable _TYPE_ is a bookkeeping variable whose value indicates that the value of _STDDEV_ is a standard value rather than an estimate.

The following statements read the value of σ_0 from DLIMITS3 and create the R chart shown in Output 46.2.2:

```
symbol color = rose;
title 'Specifying Control Limit Information';
proc shewhart data=disks2 limits=dlimits3;
   rchart time*lot / nolimit0;
run;
```

The NOLIMIT0 option suppresses the display of a fixed lower control limit if the value of the limit is zero (which is the case in this example).

Output 46.2.2. Reading in Standard Value for Process Standard Deviation



Instead of specifying σ_0 with the variable _STDDEV_ in a LIMITS= data set, you can use the SIGMA0= option in the RCHART statement. The following statements create an R chart identical to the chart shown in Output 46.2.2:

```
proc shewhart data=disks;
   rchart time*lot / sigma0=.045 nolimit0;
run;
```

For more information, see "LIMITS= Data Set" on page 1601.

Chapter 47
SCHART Statement

Chapter Contents

OVERVIEW 1613

GETTING STARTED 1614
- Creating Standard Deviation Charts from Raw Data 1614
- Creating Standard Deviation Charts from Subgroup Summary Data 1617
- Saving Summary Statistics 1619
- Saving Control Limits 1620
- Reading Preestablished Control Limits 1623

SYNTAX 1624
- Summary of Options 1625

DETAILS 1634
- Constructing Charts for Standard Deviations 1634
- Output Data Sets 1635
- ODS Tables 1639
- Input Data Sets 1639
- Methods for Estimating the Standard Deviation 1642
- Axis Labels 1644
- Missing Values 1644

EXAMPLES 1644
- Example 47.1. Specifying a Known Standard Deviation 1644
- Example 47.2. Computing Average Run Lengths for s Charts 1646

Chapter 47
SCHART Statement

Overview

The SCHART statement creates an s chart for subgroup standard deviations, which is used to analyze the variability of a process.*

You can use options in the SCHART statement to

- compute control limits from the data based on a multiple of the standard error of the plotted standard deviations or as probability limits
- tabulate subgroup sample sizes, subgroup standard deviations, control limits, and other information
- save control limits in an output data set
- save subgroup sample sizes, subgroup means, and subgroup standard deviations in an output data set
- read preestablished control limits from a data set
- specify a method for estimating the process standard deviation
- specify a known (standard) process standard deviation for computing control limits
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

*You can also use R charts for this purpose; see Chapter 46, "RCHART Statement." In general, s charts are recommended with large subgroup sample sizes ($n_i \geq 10$).

Getting Started

This section introduces the SCHART statement with simple examples that illustrate commonly used options. Complete syntax for the SCHART statement is presented in the "Syntax" section on page 1624, and advanced examples are given in the "Examples" section on page 1644.

Creating Standard Deviation Charts from Raw Data

See SHWSCHR in the SAS/QC Sample Library

A petroleum company uses a turbine to heat water into steam, which is then pumped into the ground to make oil less viscous and easier to extract. This heating process occurs 20 times daily, and the amount of power (in kilowatts) used to heat the water to the desired temperature is recorded. The following statements create a SAS data set named TURBINE, which contains the power output measurements for 20 days:

```
data turbine;
   informat day date7.;
   format day date5.;
   input day @;
   do i=1 to 10;
      input kwatts @;
      output;
      end;
   drop i;
   datalines;
 04JUL94 3196 3507 4050 3215 3583 3617 3789 3180 3505 3454
 04JUL94 3417 3199 3613 3384 3475 3316 3556 3607 3364 3721
 05JUL94 3390 3562 3413 3193 3635 3179 3348 3199 3413 3562
 05JUL94 3428 3320 3745 3426 3849 3256 3841 3575 3752 3347
 06JUL94 3478 3465 3445 3383 3684 3304 3398 3578 3348 3369
 06JUL94 3670 3614 3307 3595 3448 3304 3385 3499 3781 3711
 07JUL94 3448 3045 3446 3620 3466 3533 3590 3070 3499 3457
 07JUL94 3411 3350 3417 3629 3400 3381 3309 3608 3438 3567
 08JUL94 3568 2968 3514 3465 3175 3358 3460 3851 3845 2983
 08JUL94 3410 3274 3590 3527 3509 3284 3457 3729 3916 3633
 09JUL94 3153 3408 3741 3203 3047 3580 3571 3579 3602 3335
 09JUL94 3494 3662 3586 3628 3881 3443 3456 3593 3827 3573
 10JUL94 3594 3711 3369 3341 3611 3496 3554 3400 3295 3002
 10JUL94 3495 3368 3726 3738 3250 3632 3415 3591 3787 3478
 11JUL94 3482 3546 3196 3379 3559 3235 3549 3445 3413 3859
 11JUL94 3330 3465 3994 3362 3309 3781 3211 3550 3637 3626
 12JUL94 3152 3269 3431 3438 3575 3476 3115 3146 3731 3171
 12JUL94 3206 3140 3562 3592 3722 3421 3471 3621 3361 3370
 13JUL94 3421 3381 4040 3467 3475 3285 3619 3325 3317 3472
 13JUL94 3296 3501 3366 3492 3367 3619 3550 3263 3355 3510
 14JUL94 3795 3872 3559 3432 3322 3587 3336 3732 3451 3215
 14JUL94 3594 3410 3335 3216 3336 3638 3419 3515 3399 3709
 15JUL94 3850 3431 3460 3623 3516 3810 3671 3602 3480 3388
 15JUL94 3365 3845 3520 3708 3202 3365 3731 3840 3182 3677
 16JUL94 3711 3648 3212 3664 3281 3371 3416 3636 3701 3385
 16JUL94 3769 3586 3540 3703 3320 3323 3480 3750 3490 3395
```

```
 17JUL94 3596 3436 3757 3288 3417 3331 3475 3600 3690 3534
 17JUL94 3306 3077 3357 3528 3530 3327 3113 3812 3711 3599
 18JUL94 3428 3760 3641 3393 3182 3381 3425 3467 3451 3189
 18JUL94 3588 3484 3759 3292 3063 3442 3712 3061 3815 3339
 19JUL94 3746 3426 3320 3819 3584 3877 3779 3506 3787 3676
 19JUL94 3727 3366 3288 3684 3500 3501 3427 3508 3392 3814
 20JUL94 3676 3475 3595 3122 3429 3474 3125 3307 3467 3832
 20JUL94 3383 3114 3431 3693 3363 3486 3928 3753 3552 3524
 21JUL94 3349 3422 3674 3501 3639 3682 3354 3595 3407 3400
 21JUL94 3401 3359 3167 3524 3561 3801 3496 3476 3480 3570
 22JUL94 3618 3324 3475 3621 3376 3540 3585 3320 3256 3443
 22JUL94 3415 3445 3561 3494 3140 3090 3561 3800 3056 3536
 23JUL94 3421 3787 3454 3699 3307 3917 3292 3310 3283 3536
 23JUL94 3756 3145 3571 3331 3725 3605 3547 3421 3257 3574
;
run;
```

A partial listing of TURBINE is shown in Figure 47.1.

Kilowatt Power Output Data

Obs	day	kwatts
1	04JUL	3196
2	04JUL	3507
3	04JUL	4050
4	04JUL	3215
.	.	.
.	.	.
.	.	.
400	23JUL	3574

Figure 47.1. Partial Listing of the Data Set TURBINE

The data set TURBINE is said to be in "strung-out" form, since each observation contains the day and power output for a single heating. The first 20 observations contain the power outputs for the first day, the second 20 observations contain the power outputs for the second day, and so on. Because the variable DAY classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable KWATTS contains the power output measurements and is referred to as the *process variable* (or *process* for short).

You can use an s chart to determine whether the variability in the heating process is in control. The following statements create the s chart shown in Figure 47.2:

```
title 'Chart for Standard Deviations of Power Output';
proc shewhart data=turbine;
   schart kwatts*day;
run;
```

This example illustrates the basic form of the SCHART statement. After the keyword SCHART, you specify the *process* to analyze (in this case, KWATTS), followed by an asterisk and the *subgroup-variable* (DAY).

The input data set is specified with the DATA= option in the PROC SHEWHART statement.

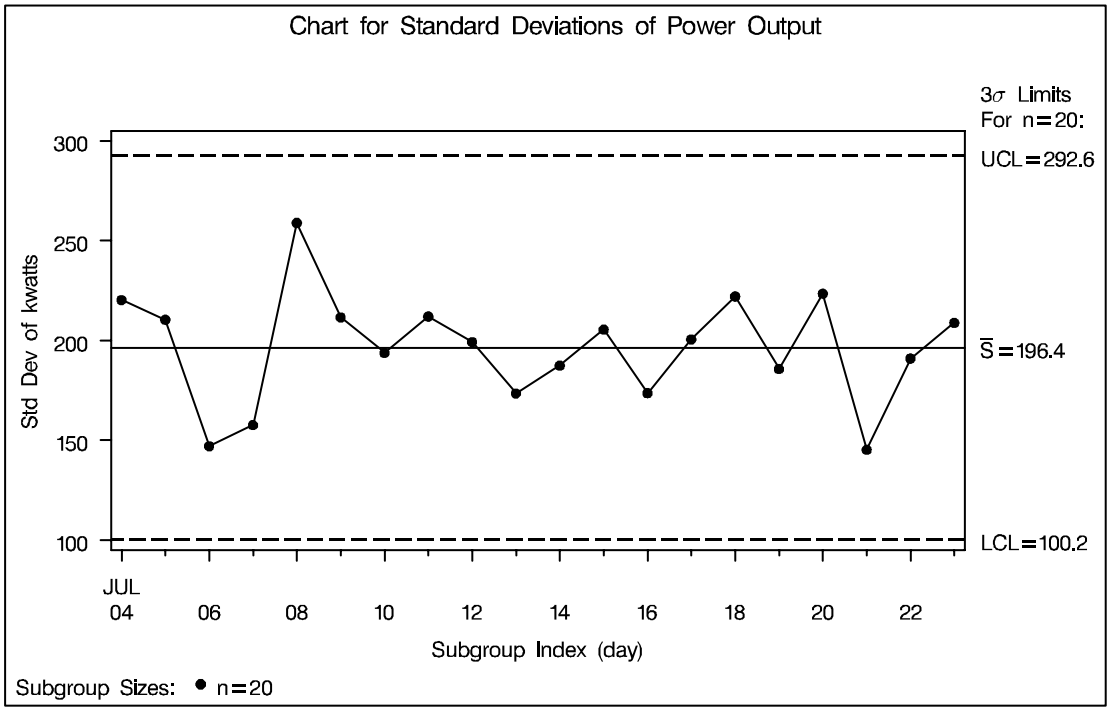


Figure 47.2. s Chart for Power Output Data

Each point on the chart represents the standard deviation of the measurements for a particular day. For instance, the standard deviation plotted for the first day is

$$\sqrt{\frac{(3196-3487.4)^2+(3507-3487.4)^2+\cdots+(3721-3487.4)^2}{19}}=220.26$$

Since all of the subgroup standard deviations lie within the control limits, you can conclude that the variability of the process is in statistical control.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 47.23 on page 1635. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1623.

For computational details, see "Constructing Charts for Standard Deviations" on page 1634. For more details on reading raw data, see "DATA= Data Set" on page 1639.

Creating Standard Deviation Charts from Subgroup Summary Data

See SHWSCHR in the SAS/QC Sample Library

The previous example illustrates how you can create s charts using raw data (process measurements). However, in many applications, the data are provided as subgroup summary statistics. This example illustrates how you can use the SCHART statement with data of this type.

The following data set (OILSUM) provides the data from the preceding example in summarized form:

```
data oilsum;
   input day kwattsx kwattss kwattsn;
   informat day date7. ;
   format day date5. ;
   label day   ='Date of Measurement';
   datalines;
 04JUL94 3487.40 220.260 20
 05JUL94 3471.65 210.427 20
 06JUL94 3488.30 147.025 20
 07JUL94 3434.20 157.637 20
 08JUL94 3475.80 258.949 20
 09JUL94 3518.10 211.566 20
 10JUL94 3492.65 193.779 20
 11JUL94 3496.40 212.024 20
 12JUL94 3398.50 199.201 20
 13JUL94 3456.05 173.455 20
 14JUL94 3493.60 187.465 20
 15JUL94 3563.30 205.472 20
 16JUL94 3519.05 173.676 20
 17JUL94 3474.20 200.576 20
 18JUL94 3443.60 222.084 20
 19JUL94 3586.35 185.724 20
 20JUL94 3486.45 223.474 20
 21JUL94 3492.90 145.267 20
 22JUL94 3432.80 190.994 20
 23JUL94 3496.90 208.858 20
;
run;
```

A partial listing of OILSUM is shown in Figure 47.3. There is exactly one observation for each subgroup (note that the subgroups are still indexed by DAY). The variable KWATTSX contains the subgroup means, the variable KWATTSS contains the subgroup standard deviations, and the variable KWATTSN contains the subgroup sample sizes (these are all 20).

Summary Data Set for Power Outputs

day	kwattsx	kwattss	kwattsn
04JUL	3487.40	220.260	20
05JUL	3471.65	210.427	20
06JUL	3488.30	147.025	20
.	.	.	.
.	.	.	.
.	.	.	.

Figure 47.3. The Summary Data Set OILSUM

You can read this data set by specifying it as a HISTORY= data set in the PROC SHEWHART statement, as follows:

```
title 'Chart for Standard Deviations of Power Output';
proc shewhart history=oilsum lineprinter;
   schart kwatts*day='*';
run;
```

The resulting s chart is shown in Figure 47.4. Since the LINEPRINTER option is specified in the PROC SHEWHART statement, line printer output is produced. The asterisk (*) specified in single quotes after the *subgroup-variable* indicates the character used to plot points. This character must follow an equal sign.

Note that KWATTS is *not* the name of a SAS variable in the data set OILSUM but is, instead, the common prefix for the names of the SAS variables KWATTSS and KWATTSN. The suffix characters S and N indicate *standard deviation* and *sample size*, respectively. Thus, you can specify two subgroup summary variables in the HISTORY= data set with a single name (KWATTS), which is referred to as the *process*. The name DAY, specified after the asterisk, is the name of the *subgroup-variable*.

In general, a HISTORY= input data set used with the SCHART statement must contain the following variables:

- subgroup variable
- subgroup standard deviation variable
- subgroup sample size variable

Furthermore, the names of the subgroup standard deviation and sample size variables must begin with the *process* name specified in the SCHART statement and end with the special suffix characters S and N, respectively. If the names do not follow this convention, you can use the RENAME option in the PROC SHEWHART statement to rename the variables for the duration of the SHEWHART procedure step (see page 1743).

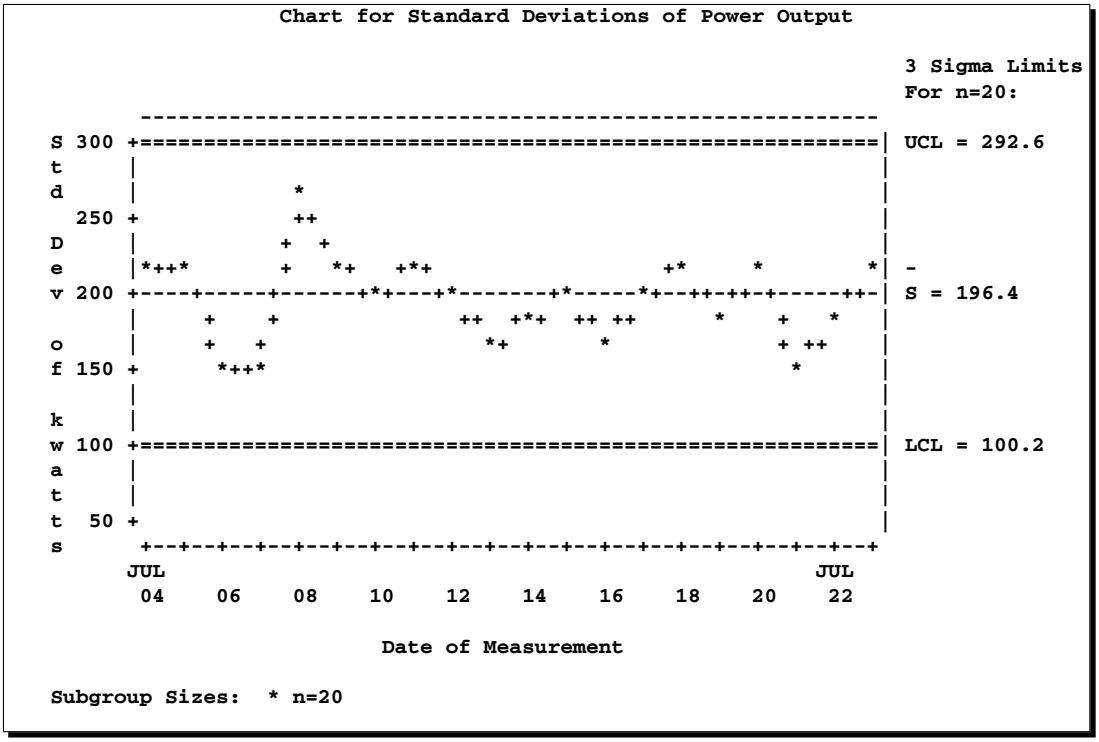


Figure 47.4. *s* Chart for Power Output Data

In summary, the interpretation of *process* depends on the input data set.

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names of the variables containing the summary statistics.

For more information, see "HISTORY= Data Set" on page 1640.

Saving Summary Statistics

In this example, the SCHART statement is used to create a summary data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read measurements from the data set TURBINE and create a summary data set named TURBHIST:

See SHWSCHR in the SAS/QC Sample Library

```
proc shewhart data=turbine;
   schart kwatts*day / outhistory = turbhist
                       nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 47.2. Options such as OUTHISTORY= and NOCHART are specified after the slash (/) in the SCHART statement. A complete list of options is presented in the "Syntax" section on page 1624.

Figure 47.5 contains a partial listing of TURBHIST.

Summary Data Set for Power Output

day	kwattsX	kwattsS	kwatts N
04JUL	3487.40	220.260	20
05JUL	3471.65	210.427	20
06JUL	3488.30	147.025	20
07JUL	3434.20	157.637	20
08JUL	3475.80	258.949	20
.	.	.	.
.	.	.	.
.	.	.	.

Figure 47.5. The Summary Data Set TURBHIST

There are four variables in the data set TURBHIST.

- DAY contains the subgroup index.
- KWATTSX contains the subgroup means.
- KWATTSS contains the subgroup standard deviations.
- KWATTSN contains the subgroup sample sizes.

The subgroup mean variable is included even though it is not required by the SCHART statement. This allows the data set to be used as a HISTORY= data set with the BOXCHART, XCHART, and XSCHART statements, as well as with the SCHART statement. Note that the summary statistic variables are named by adding the suffix characters *X*, *S*, and *N* to the *process* KWATTS specified in the SCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 1637.

Saving Control Limits

See SHWSCHR in the SAS/QC Sample Library

You can save the control limits for an s chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1623) or modify the limits with a DATA step program.

The following statements read measurements from the data set TURBINE (see page 1614) and save the control limits displayed in Figure 47.2 in a data set named TURBLIM:

```
proc shewhart data=turbine;
   schart kwatts*day / outlimits=turblim
                       nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. The data set TURBLIM is listed in Figure 47.6.

Control Limits for Power Output Data

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_
kwatts	day	ESTIMATE	20	.002792725	3	3351.92

MEAN	_UCLX_	_LCLS_	_S_	_UCLS_	_STDDEV_
3485.41	3618.90	100.207	196.396	292.584	198.996

Figure 47.6. The Data Set TURBLIM Containing Control Limit Information

The data set TURBLIM contains one observation with the limits for *process* KWATTS. The variables _LCLS_ and _UCLS_ contain the lower and upper control limits, and the variable _S_ contains the central line. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values. The variables _LCLX_ and _UCLX_, which contain the lower and upper control limits for subgroup means, are included so that the data set TURBLIM can be used to create an $\bar{X}$ chart (see Chapter 51, "XSCHART Statement,"). For more information, see "OUTLIMITS= Data Set" on page 1635.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=turbine;
   schart kwatts*day / outtable=turbtab
                       nochart;
run;
```

The data set TURBTAB is listed in Figure 47.7.

This data set contains one observation for each subgroup sample. The variables _SUBS_ and _SUBN_ contain the subgroup standard deviations and subgroup sample sizes. The variables _LCLS_ and _UCLS_ contain the lower and upper control limits, and the variable _S_ contains the central line. The variables _VAR_ and BATCH contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1638.

Summary Statistics and Control Limit Information

VAR	day	_SIGMAS_	_LIMITN_	_SUBN_	_LCLS_	_SUBS_	_S_	_UCLS_	_EXLIM_
kwatts	04JUL	3	20	20	100.207	220.260	196.396	292.584	
kwatts	05JUL	3	20	20	100.207	210.427	196.396	292.584	
kwatts	06JUL	3	20	20	100.207	147.025	196.396	292.584	
kwatts	07JUL	3	20	20	100.207	157.637	196.396	292.584	
kwatts	08JUL	3	20	20	100.207	258.949	196.396	292.584	
kwatts	09JUL	3	20	20	100.207	211.566	196.396	292.584	
kwatts	10JUL	3	20	20	100.207	193.779	196.396	292.584	
kwatts	11JUL	3	20	20	100.207	212.024	196.396	292.584	
kwatts	12JUL	3	20	20	100.207	199.201	196.396	292.584	
kwatts	13JUL	3	20	20	100.207	173.455	196.396	292.584	
kwatts	14JUL	3	20	20	100.207	187.465	196.396	292.584	
kwatts	15JUL	3	20	20	100.207	205.472	196.396	292.584	
kwatts	16JUL	3	20	20	100.207	173.676	196.396	292.584	
kwatts	17JUL	3	20	20	100.207	200.576	196.396	292.584	
kwatts	18JUL	3	20	20	100.207	222.084	196.396	292.584	
kwatts	19JUL	3	20	20	100.207	185.724	196.396	292.584	
kwatts	20JUL	3	20	20	100.207	223.474	196.396	292.584	
kwatts	21JUL	3	20	20	100.207	145.267	196.396	292.584	
kwatts	22JUL	3	20	20	100.207	190.994	196.396	292.584	
kwatts	23JUL	3	20	20	100.207	208.858	196.396	292.584	

Figure 47.7. The OUTTABLE= Data Set TURBTAB

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read TURBTAB and display an s chart (not shown here) identical to the chart in Figure 47.2:

```
title 'Chart for Standard Deviations of Power Output';
symbol v=dot;
proc shewhart table=turbtab;
   schart kwatts*day;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see Chapter 56, "Specialized Control Charts,"). For more information, see "TABLE= Data Set" on page 1641.

Reading Preestablished Control Limits

See SHWSCHR in the SAS/QC Sample Library

In the previous example, the OUTLIMITS= data set TURBLIM saved control limits computed from the measurements in TURBINE. This example shows how these limits can be applied to new data.

The following statements create an s chart for new measurements in the data set TURBINE2 (not listed here) using the control limits in TURBLIM:

```
title 'Chart for Standard Deviations of Power Output';
proc shewhart data=turbine2 limits=turblim;
   schart kwatts*day;
run;
```

The chart is shown in Figure 47.8. The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default,* this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name KWATTS
- the value of _SUBGRP_ matches the *subgroup-variable* name DAY

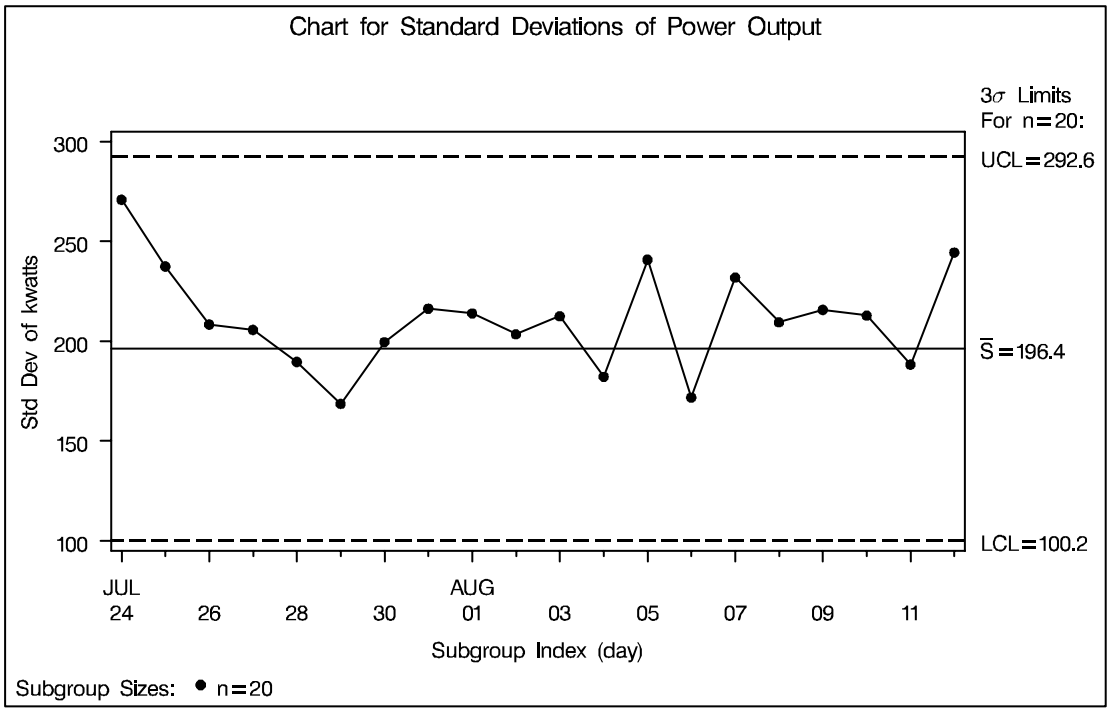


Figure 47.8. s Chart for Second Set of Power Output Data

*In Release 6.09 and in earlier releases, it is also necessary to specify the READLIMITS option to read control limits from a LIMITS= data set.

All the standard deviations lie within the control limits, indicating that the variability of the heating process is still in statistical control.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1640 for details concerning the variables that you must provide.

Syntax

The basic syntax for the SCHART statement is as follows:

SCHART *process*******subgroup-variable* **;**

The general form of this syntax is as follows:

SCHART (*processes***)****subgroup-variable* <**(***block-variables* **)** >
< =*symbol-variable* | =*'character'* > < **/** *options* >**;**

You can use any number of SCHART statements in the SHEWHART procedure. The components of the SCHART statement are described as follows.

process
processes
identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating Standard Deviation Charts from Raw Data" on page 1614.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating Standard Deviation Charts from Subgroup Summary Data" on page 1617.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1620.

A *process* is required. If you specify more than one *process*, enclose the list in parentheses. For example, the following statements request distinct s charts for WEIGHT, LENGTH, and WIDTH:

```
proc shewhart data=measures;
   schart (weight length width)*day;
run;
```

subgroup-variable
is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding SCHART statement, DAY is the subgroup variable. For details, see "Subgroup Variables" on page 1771.

block-variables
are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 1932 for an example.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the subgroup standard deviations.

- If you produce a chart on a line printer, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce a chart on a graphics device, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOLn statements. See "Displaying Stratification in Levels of a Classification Variable" on page 1931 for an example.

character
specifies a plotting character for charts produced on line printers. For example, the following statements create an s chart using an asterisk (*) to plot the points:

```
proc shewhart data=values;
   schart weight*day='*';
run;
```

options
enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The "Summary of Options" section, which follows, lists all options by function. Chapter 53, "Dictionary of Options," describes each option in detail.

Summary of Options

The following tables list the SCHART statement options by function. For complete descriptions, see Chapter 53, "Dictionary of Options."

Table 47.1. Tabulation Options

TABLE	creates a basic table of subgroup standard deviations, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with a column indicating which tests for special causes are positive

Note that specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.

Table 47.2. Options for Specifying Tests for Special Causes

NO3SIGMACHECK	allows tests to be applied with control limits other than 3σ limits
TESTS2=*value-list*| *customized-pattern-list*	specifies tests for special causes for the s chart
TEST2RESET=*variable*	allows tests for special causes to be reset for the s chart
TEST2RUN=*n*	specifies length of pattern for Test 2
TEST3RUN=*n*	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL='*label*'| (*variable*)|*keyword*	provides labels for points where test is positive
TESTLABEL*n*='*label*'	specifies label for n^{th} test for special causes
TESTNMETHOD= STANDARDIZE	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
ZONE2LABELS	adds labels A, B, and C to zone lines for s chart
ZONE2VALUES	labels zone lines with their values
ZONES2	adds lines to s chart delineating zones A, B, and C
ZONEVALPOS=*n*	specifies position of ZONE2VALUES labels

Table 47.3. Graphical Options for Displaying Tests for Special Causes

CTESTLABBOX=*color*	specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=*color*| *test-color-list*	specifies color for labels indicating points where test is positive
CTESTSYMBOL=*color*	specifies color for symbol used to plot points where test is positive
CZONES=*color*	specifies color for lines and labels delineating zones A, B, and C
LTESTS=*linetype*	specifies type of line connecting points where test is positive
LZONES=*linetype*	specifies line type for lines delineating zones A, B, and C
TESTFONT=*font*	specifies software font for labels at points where test is positive
TESTHEIGHT=*value*	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=*symbol*	specifies plot symbol for points where test is positive
TESTSYMBOLHT=*value*	specifies symbol height for points where test is positive
WTESTS=*n*	specifies width of line connecting points where test is positive

Table 47.4. Line Printer Options for Displaying Tests for Special Causes

TESTCHAR='*character*'	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
ZONECHAR='*character*'	specifies character for lines that delineate zones for tests for special causes

Table 47.5. Clipping Options

CCLIP=*color*	specifies color for plot symbol for clipped points
CLIPCHAR='*character*'	specifies plot character for clipped points
CLIPFACTOR=*value*	determines extent to which extreme points are clipped
CLIPLEGEND='*string*'	specifies text for clipping legend
CLIPLEGPOS=*keyword*	specifies position of clipping legend
CLIPSUBCHAR='*character*'	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=*symbol*	specifies plot symbol for clipped points
CLIPSYMBOLHT=*value*	specifies symbol marker height for clipped points

Table 47.6. Reference Line Options

CHREF=*color*	specifies color for lines requested by HREF= option
CVREF=*color*	specifies color for lines requested by VREF= option
HREF=*values*|*SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on *s* chart
HREFCHAR='*character*'	specifies line character for HREF= lines
HREFDATA=*SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on individual measurements chart
HREFLABELS='*label1*'...'*labeln*'	specifies labels for HREF= lines
HREFLABPOS=*n*	specifies position of HREFLABELS= labels
LHREF=*linetype*	specifies line type for HREF= lines
LVREF=*linetype*	specifies line type for VREF= lines
NOBYREF	specifies that reference line information in a data set is to be applied uniformly to charts created for all BY groups
VREF=*values*|*SAS-data-set*	specifies position of reference lines perpendicular to vertical axis on *s* chart
VREFCHAR='*character*'	specifies line character for VREF= lines
VREFLABELS='*label1*'...'*labeln*'	specifies labels for VREF= lines
VREFLABPOS=*n*	specifies position of VREFLABELS= labels

Table 47.7. Block Variable Legend Options

BLOCKLABELPOS=*keyword*	specifies position of label for *block-variable* legend
BLOCKLABTYPE=*value*|*keyword*	specifies text size of *block-variable* legend
BLOCKPOS=*n*	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=*color-list*	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=*variable*|(*variables*)	specifies one or more variables whose values are colors for filling background of *block-variable* legend

Table 47.8. Axis and Axis Label Options

CAXIS=*color*	specifies color for axis lines and tick marks
CFRAME=*color*|(*color-list*)	specifies fill colors for frame for plot area
CTEXT=*color*	specifies color for tick mark values and axis labels
HAXIS=*values*|AXISn	specifies major tick mark values for horizontal axis
HEIGHT=*value*	specifies height of axis label and axis legend text
HMINOR=*n*	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=*value*	specifies length of offset at both ends of horizontal axis
INTSTART=*value*	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	suppresses vertical axis truncation at zero applied by default to s chart
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=*n*	specifies thinning factor for tick mark labels on horizontal axis
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=*values*|AXISn	specifies major tick mark values for vertical axis on s chart
VFORMAT=*format*	specifies format for vertical axis tick mark labels
VMINOR=*n*	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=*value*	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis
WAXIS=*n*	specifies width of axis lines

Table 47.9. Plot Layout Options

ALLN	plots summary statistics for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts for a process variable only when exceptions occur
INTERVAL=*keyword*	specifies natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=*n*	specifies maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of s chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=*n*	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=*n*	specifies number of pages or screens to be used to display chart
ZEROSTD	displays s chart regardless of whether $\hat{\sigma} = 0$

Table 47.10. Options for Specifying Control Limits

ALPHA=*value*	requests probability limits for control charts
LIMITN=*n*|VARYING	specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	computes control limits for each *process* from the data rather than from a LIMITS= data set (Release 6.10 and later releases)
READALPHA	reads the variable _ALPHA_ instead of the variable _SIGMAS_ from a LIMITS= data set
READINDEXES=ALL| '*label1*'...'*labeln*'	reads multiple sets of control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (Release 6.09 and earlier releases)
SIGMAS=*k*	specifies width of control limits in terms of multiple k of standard error of plotted statistic

Table 47.11. Options for Displaying Control Limits

CINFILL=*color*	specifies color for area inside control limits
CLIMITS=*color*	specifies color of control limits, central line, and related labels
LCLLABEL='*label*'	specifies label for lower control limit on s chart
LIMLABSUBCHAR='*character*'	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=*linetype*	specifies line type for control limits
NDECIMAL=*n*	specifies number of digits to right of decimal place in default labels for control limits and central line on s chart
NOCTL	suppresses display of central line on s chart
NOLCL	suppresses display of lower control limit on s chart
NOLIMITLABEL	suppresses labels for control limits and central line
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOLIMIT0	suppresses display of zero lower control limit on s chart
NOUCL	suppresses display of upper control limit on s chart
SSYMBOL='*string*'|*keyword*	specifies label for central line on s chart
UCLLABEL='*string*'	specifies label for upper control limit on s chart
WLIMITS=*n*	specifies width for control limits and central line

Table 47.12. Grid Options

CGRID=*color*	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=*linetype*	specifies line type for grid requested with the ENDGRID option
LGRID=*linetype*	specifies line type for grid requested with the GRID option
WGRID=*n*	specifies width of grid lines

Table 47.13. Options for Plotting and Labeling Points

ALLLABEL2=VALUE|(*variable*)	labels every point on s chart
CLABEL=*color*	specifies color for labels
CCONNECT=*color*	specifies color for line segments that connect points on chart
CFRAMELAB=*color*	specifies fill color for frame around labeled points
CNEEDLES=*color*	specifies color for needles that connect points to central line
CONNECTCHAR='*character*'	specifies character used to form line segments that connect points on chart
COUT=*color*	specifies color for portions of line segments that connect points outside control limits
COUTFILL=*color*	specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=*angle*	specifies angle at which labels are drawn
LABELFONT=*font*	specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=*value*	specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on chart
OUTLABEL2=VALUE|(*variable*)	labels points outside control limits on s chart
SYMBOLCHARS='*characters*'	specifies characters indicating *symbol-variable*
SYMBOLLEGEND=NONE|*name*	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=*keyword*	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=*n*	specifies width of needles

Table 47.14. Input Data Set Options

MISSBREAK	specifies that observations with missing values are not to be processed

Table 47.15. Output Data Set Options

OUTHISTORY=*SAS-data-set*	creates output data set containing subgroup summary statistics
OUTINDEX='*string*'	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=*SAS-data-set*	creates output data set containing control limits
OUTTABLE=*SAS-data-set*	creates output data set containing subgroup summary statistics and control limits

Table 47.16. Phase Options

CPHASELEG=*color*	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE='*string*'	specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=*value*| *keyword*	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES= ALL| '*label1*'...'*labeln*'	specifies *phases* to be read from an input data set

Table 47.17. Specification Limit Options

CIINDICES (ALPHA=*value* TYPE=*keyword*)	specifies α value and type for computing capability index confidence limits
LSL=*value-list*	specifies list of lower specification limits
TARGET=*value-list*	specifies list of target values
USL=*value-list*	specifies list of upper specification limits

Table 47.18. Process Mean and Standard Deviation Options

SIGMA0=*value*	specifies known value σ_0 for process standard deviation σ
SMETHOD=*keyword*	specifies method for estimating process standard deviation σ
TYPE=*keyword*	identifies whether parameters are estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set

Table 47.19. Graphical Enhancement Options

ANNOTATE= *SAS-data-set*	specifies annotate data set that adds features to s chart
DESCRIPTION='*string*'	specifies string that appears in the description field of PROC GREPLAY master menu for s chart
FONT=*font*	specifies software font for labels and legends on charts
NAME='*string*'	specifies name that appears in the name field of the PROC GREPLAY master menu for s chart
PAGENUM='*string*'	specifies the form of the label used in pagination
PAGENUMPOS= *keyword*	specifies the position of the page number requested with the PAGENUM= option

Table 47.20. Star Options

CSTARCIRCLES=*color*	specifies color for STARCIRCLES= circles
CSTARFILL=*color*| *(variable)*	specifies color for filling stars
CSTAROUT=*color*	specifies outline color for stars exceeding inner or outer circles
CSTARS=*color*| *(variable)*	specifies color for outlines of stars
LSTARCIRCLES= *linetypes*	specifies line types for STARCIRCLES= circles
LSTARS=*linetype*|*(variable)*	specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=*value*	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=*value-list*	specifies reference circles for stars
STARINRADIUS=*value*	specifies inner radius of stars
STARLABEL=*keyword*	specifies vertices to be labeled
STARLEGEND=*keyword*	specifies style of legend for star vertices
STARLEGENDLAB=*'label'*	specifies label for STARLEGEND= legend
STAROUTRADIUS=*value*	specifies outer radius of stars
STARSPEC=*value*| *SAS-data-set*	specifies method used to standardize vertex variables
STARSTART=*value*	specifies angle for first vertex
STARTYPE=*keyword*	specifies graphical style of star
STARVERTICES= *variable*|*(variables)*	superimposes star at each point on s chart
WSTARCIRCLES=*n*	specifies width of STARCIRCLES= circles
WSTARS=*n*	specifies width of STARVERTICES= stars

Table 47.21. Overlay Options

CCOVERLAY=*color-list*	specifies colors for overlay line segments
COVERLAY=*color-list*	specifies colors for overlay plots
COVERLAYCLIP=*color*	specifies color for clipped points on overlays
LOVERLAY=*linetypes*	specifies line types for overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=*variable-list*	specifies variables to overlay on control chart
OVERLAYCLIPSYM=*symbol*	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=*value*	specifies symbol height for clipped points on overlays
OVERLAYHTML=*variable-list*	specifies URLS to associate with overlay points
OVERLAYID=*variable-list*	specifies labels for primary chart overlay points
OVERLAYLEGLAB='*label*'	specifies label for overlay legend
OVERLAYSYM=*symbol-list*	specifies symbols for overlay plots
OVERLAYSYMHT=*value-list*	specifies symbol heights for overlay plots
WOVERLAY=*value-list*	specifies widths of overlay line segments

Table 47.22. Options for Interactive Control Charts

HTML=(*variable*)	specifies a variable whose values are URLs to be associated with subgroups
HTML_LEGEND=(*variable*)	specifies a variable whose values are URLs to be associated with symbols in the symbol legend
WEBOUT=*SAS-data-set*	creates an OUTTABLE= data set with additional graphics coordinate data

Details

Constructing Charts for Standard Deviations

The following notation is used in this section:

σ	process standard deviation (standard deviation of the population of measurements)
s_i	standard deviation of measurements in i^{th} subgroup $$s_i = \sqrt{(1/(n_i-1))((x_{i1}-\bar{X}_i)^2+\cdots+(x_{in_i}-\bar{X}_i)^2)}$$
n_i	sample size of i^{th} subgroup
$c_4(n)$	expected value of the standard deviation of n independent normally distributed variables with unit standard deviation
$c_5(n)$	standard error of the standard deviation of n independent observations from a normal population with unit standard deviation
$\chi^2_p(n)$	$100p^{\text{th}}$ percentile $(0 < p < 1)$ of the χ^2 distribution with n degrees of freedom

Plotted Points

Each point on an s chart indicates the value of a subgroup standard deviation (s_i). For example, if the tenth subgroup contains the values 12, 15, 19, 16, and 13, the value plotted for this subgroup is

$$s_{10} = \sqrt{((12-15)^2+(15-15)^2+(19-15)^2+(16-15)^2+(13-15)^2)/4} = 2.739$$

Central Line

By default, the central line for the i^{th} subgroup indicates an estimate for the expected value of s_i, which is computed as $c_4(n_i)\hat{\sigma}$, where $\hat{\sigma}$ is an estimate of σ. If you specify a known value (σ_0) for σ, the central line indicates the value of $c_4(n_i)\sigma_0$. Note that the central line varies with n_i.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of s_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that s_i exceeds the limits

The following table provides the formulas for the limits:

Table 47.23. Limits for s Charts

Control Limits
LCL = lower limit = $\max(c_4(n_i)\hat{\sigma} - kc_5(n_i)\hat{\sigma}, 0)$ UCL = upper limit = $c_4(n_i)\hat{\sigma} + kc_5(n_i)\hat{\sigma}$

Probability Limits
LCL = lower limit = $\hat{\sigma}\sqrt{\chi^2_{\alpha/2}(n_i - 1)/(n_i - 1)}$ UCL = upper limit = $\hat{\sigma}\sqrt{\chi^2_{1-\alpha/2}(n_i - 1)/(n_i - 1)}$

The formulas assume that the data are normally distributed. If a standard value σ_0 is available for σ, replace $\hat{\sigma}$ with σ_0 in Table 47.23. Note that the upper and lower limits vary with n_i and that the probability limits are asymmetric around the central line.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables are saved:

Table 47.24. OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
_CP_	capability index C_p
CPK	capability index C_{pk}
CPL	capability index CPL
CPM	capability index C_{pm}
CPU	capability index CPU
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLS	lower control limit for subgroup standard deviation
LCLX	lower control limit for subgroup mean
LIMITN	sample size associated with the control limits
LSL	lower specification limit
MEAN	process mean ($\overline{\overline{X}}$ or μ_0)
_S_	value of central line on s chart
SIGMAS	multiple (k) of standard error of $\bar{X}_i$ or s_i
STDDEV	process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*subgroup-variable* specified in the SCHART statement
TARGET	target value
TYPE	type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLS	upper control limit for subgroup standard deviation
UCLX	upper control limit for subgroup mean
USL	upper specification limit
VAR	*process* specified in the SCHART statement

Notes:

1. The variables _LCLX_, _MEAN_, and _UCLX_ are saved to allow the OUTLIMITS= data set to be used as a LIMITS= data set with the BOXCHART, XCHART, and XSCHART statements.
2. If the control limits vary with subgroup sample size, the special missing value *V* is assigned to the variables _LIMITN_, _LCLX_, _UCLX_, _LCLS_, _S_, and _UCLS_.
3. If the limits are defined in terms of a multiple k of the standard error of s_i, the value of _ALPHA_ is computed as

$$F_S(\text{\_LCLS\_}/\text{\_STDDEV\_}) + 1 - F_S(\text{\_UCLS\_}/\text{\_STDDEV\_})$$

   where $F_S(\cdot)$ is the cumulative distribution function of the standard deviation of a sample of n observations from a normal population with unit standard deviation, and n is the value of _LIMITN_. If _LIMITN_ has the special missing value *V*, this value is assigned to _ALPHA_.
4. If the limits are probability limits, the value of _SIGMAS_ is computed as $(\text{\_UCLS\_} - \text{\_S\_})/e$, where e is the standard error of the standard deviation of n observations from a normal population with unit standard devia-

tion. If _LIMITN_ has the special missing value *V*, this value is assigned to _SIGMAS_.

5. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you provide a target value with the TARGET= option. See "Capability Indices" on page 1774 for computational details.
6. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the SCHART statement. For an example, see "Saving Control Limits" on page 1620.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables are saved:

- the *subgroup-variable*
- a subgroup mean variable named by *process* suffixed with *X*
- a subgroup standard deviation variable named by *process* suffixed with *S*
- a subgroup sample size variable named by *process* suffixed with *N*

The subgroup mean variable is included so that the data set can be reused as a HISTORY= data set with the BOXCHART, XCHART, and XSCHART statements, as well as the SCHART statement.

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the SCHART statement. For example, consider the following statements:

```
proc shewhart data=steel;
   schart (width diameter)*lot / outhistory=summary;
run;
```

The data set SUMMARY contains variables named LOT, WIDTHX, WIDTHS, WIDTHN, DIAMTERX, DIAMTERS, and DIAMTERN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 1619.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables are saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on s chart
LCLS	lower control limit for standard deviation
LIMITN	nominal sample size associated with the control limits
_S_	average standard deviation
SIGMAS	multiple (k) of the standard error associated with control limits
subgroup	values of the subgroup variable
SUBN	subgroup sample size
SUBS	subgroup standard deviation
TESTS2	tests for special causes signaled on s chart
UCLS	upper control limit for standard deviation
VAR	*process* specified in the SCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= option, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS2_ is saved if you specify the TESTS2= option.
3. The variables _EXLIM_ and _TESTS2_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1620.

ODS Tables

The following table summarizes the ODS tables that you can request with the SCHART statement.

Table 47.25. ODS Tables Produced with the SCHART Statement

Table Name	Description	Options
SCHART	s chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the SCHART statement must be a SAS variable in the DATA= data set. This variable provides measurements, which must be grouped into subgroup samples indexed by the values of the *subgroup-variable*. The *subgroup-variable*, which is specified in the SCHART statement, must also be a SAS variable in the DATA= data set.

Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. If the i^{th} subgroup contains n_i items, there should be n_i consecutive observations for which the value of the *subgroup-variable* is the index of the i^{th} subgroup. For example, if each subgroup contains five items and there are 30 subgroup samples, the DATA= data set should contain 150 observations. Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the DATA= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 1936).

For an example of a DATA= data set, see "Creating Standard Deviation Charts from Raw Data" on page 1614.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set CONLIMS:*

```
proc shewhart data=info limits=conlims;
   schart weight*batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set; see Table 47.24 on page 1636. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLS_, _S_, and _UCLS_, which specify the control limits directly
- the variable _STDDEV_, which is used to calculate the control limits according to the equations in Table 47.23 on page 1635

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option. This must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are `ESTIMATE`, `ESTIMATE`, `STDMU`, and `STDSIGMA`.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1623.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This allows you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART, CUSUM, or MACONTROL procedures or to read output data sets created with SAS summarization procedures, such as PROC MEANS.

*In Release 6.09 and in earlier releases, it is necessary to specify the READLIMITS option.

A HISTORY= data set used with the SCHART statement must contain the following:

- the *subgroup-variable*
- a subgroup standard deviation variable for each *process*
- a subgroup sample size variable for each *process*

The names of the subgroup standard deviation and subgroup sample size variables must be the *process* name concatenated with the special suffix characters S and N, respectively. For example, consider the following statements:

```
proc shewhart history=summary;
   schart (weight yldstren)*batch;
run;
```

The data set SUMMARY must include the variables BATCH, WEIGHTS, WEIGHTN, YLDSRENS, and YLDSRENN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 1936 for an example).

For an example of a HISTORY= data set, see "Creating Standard Deviation Charts from Subgroup Summary Data" on page 1617.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in Chapter 56, "Specialized Control Charts."

The following table lists the variables required in a TABLE= data set used with the SCHART statement:

Table 47.26. Variables Required in a TABLE= Data Set

Variable	Description
LCLS	lower control limit for standard deviation
LIMITN	nominal sample size associated with the control limits
_S_	average standard deviation
subgroup-variable	values of the *subgroup-variable*
SUBN	subgroup sample size
SUBS	subgroup standard deviation
UCLS	upper control limit for standard deviation

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS2_ (if the TESTS2= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1620.

Methods for Estimating the Standard Deviation

When control limits are determined from the input data, three methods (referred to as default, MVLUE, and RMSDF) are available for estimating σ.

Default Method

The default estimate for σ is

$$\hat{\sigma} = \frac{s_1/c_4(n_1) + \cdots + s_N/c_4(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, s_i is the sample standard deviation of the i^{th} subgroup

$$s_i = \sqrt{\frac{1}{n_i - 1}\sum_{j=1}^{n_i}(x_{ij} - \bar{X}_i)^2}$$

and

$$c_4(n_i) = \frac{\Gamma(n_i/2)\sqrt{2/(n_i - 1)}}{\Gamma((n_i - 1)/2)}$$

Here $\Gamma(\cdot)$ denotes the gamma function, and $\bar{X}_i$ denotes the i^{th} subgroup mean. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$. If the observations are normally distributed, then the expected value of s_i is $c_4(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the *ASTM Manual on Presentation of Data and Control Chart Analysis* (1976).

MVLUE Method

If you specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). This estimate is a weighted average of N unbiased estimates of σ of the form $s_i/c_4(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{h_1 s_1/c_4(n_1) + \cdots + h_N s_N/c_4(n_N)}{h_1 + \cdots + h_N}$$

where

$$h_i = \frac{[c_4(n_i)]^2}{1 - [c_4(n_i)]^2}$$

A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

RMSDF Method

If you specify SMETHOD=RMSDF, a weighted root-mean-square estimate is computed for σ.

$$\hat{\sigma} = \frac{\sqrt{(n_1 - 1)s_1^2 + \cdots + (n_N - 1)s_N^2}}{c_4(n)\sqrt{n_1 + \cdots + n_N - N}}$$

The weights are the degrees of freedom $n_i - 1$. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$.

If the unknown standard deviation σ is constant across subgroups, the root-mean-square estimate is more efficient than the minimum variance linear unbiased estimate. However, in process control applications, it is generally not assumed that σ is constant, and if σ varies across subgroups, the root-mean-square estimate tends to be more inflated than the MVLUE.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical	DATA=	*process*
Vertical	HISTORY=	subgroup standard deviation variable
Vertical	TABLE=	_SUBS_

For an example, see "Labeling Axes" on page 1966.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples

This section provides advanced examples of the SCHART statement.

Example 47.1. Specifying a Known Standard Deviation

See SHWSEX1 in the SAS/QC Sample Library

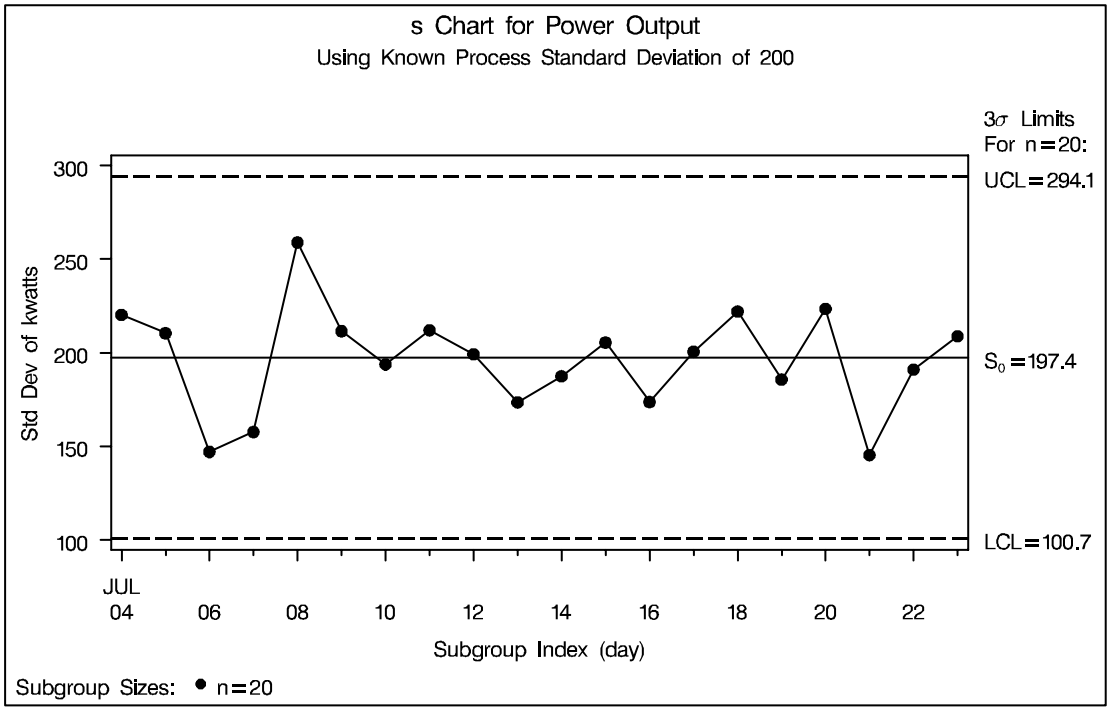
In some applications, a standard value σ_0 may be available for the process standard deviation σ. This example shows how you can specify σ_0 to compute the control limits.

Suppose that the amount of power needed to heat water in the heating process described on page 1614 has a known standard deviation of 200. The following statements specify this known value and create an s chart, shown in Output 47.1.1, for the power output measurements in the data set TURBINE:

```
title 's Chart for Power Output';
title2 'Using Known Process Standard Deviation of 200';
proc shewhart data=turbine;
   schart kwatts*day / sigma0  = 200
                       ssymbol = s0;
run;
```

The SIGMA0= option specifies σ_0, and the SSYMBOL= option specifies a label for the central line indicating that the central line is computed from σ_0. Since all the points lie within the limits, you can conclude that the variability of the process is stable.

Output 47.1.1. Reading in Standard Value for Process Standard Deviation



You can also specify σ_0 as the value of the variable _STDDEV_ in a LIMITS= data set, as illustrated by the following statements:*

```
data plimits;
   length _var_ _subgrp_ _type_ $8;
   _var_    = 'kwatts';
   _subgrp_ = 'day';
   _type_   = 'STDSIGMA';
   _limitn_ = 20;
   _stddev_ = 200;
run;

title 'Chart Using Known Process Standard Deviation';
symbol v=dot;
proc shewhart data=turbine limits=plimits;
   schart kwatts*day / ssymbol=s0;
run;
```

The resulting s chart (not shown here) is identical to the one shown in Output 47.1.1. For more information, see "LIMITS= Data Set" on page 1640.

*In Release 6.09 and in earlier releases, it is necessary to specify the READLIMITS option.

Example 47.2. Computing Average Run Lengths for s Charts

See SHWSARL in the SAS/QC Sample Library

This example illustrates how you can compute the average run length of an s chart. The data used here are the power measurements in the data set TURBINE, which is introduced on page 1614.

The in-control average run length of a Shewhart chart is ARL $= \frac{1}{p}$, where p is the probability that a single point exceeds its control limits. Since this probability is saved as the value of the variable _ALPHA_ in an OUTLIMITS= data set, you can compute ARL for an s chart as follows:

```
title 'Average In-Control Run Length';
proc shewhart data=turbine;
   schart kwatts*day / outlimits=turblim nochart;

data arlcomp;
   keep _var_ _sigmas_ _alpha_ arl;
   set turblim;
   arl = 1 / _alpha_;
run;
```

The data set ARLCOMP is listed in Output 47.2.1, which shows that the ARL is equal to 358.

Output 47.2.1. The Data Set ARLCOMP

Average In-Control Run Length

VAR	_ALPHA_	_SIGMAS_	arl
kwatts	.002792725	3	358.073

To compute out-of-control average run lengths, define f as the slippage factor for the process standard deviation σ, where $f > 1$. In other words, the "shifted" standard deviation to be detected by the chart is $f\sigma$. The following statements compute the ARL as a function of f:

```
data arlshift;
   keep f f_std p arl_f;
   set turblim;
   df = _limitn_ - 1;
   do f = 1 to 1.5 by 0.05;
      f_std = f * _stddev_;
      low   = df * ( _lcls_ / f_std )**2;
      upp   = df * ( _ucls_ / f_std )**2;
      p     = probchi( low, df ) + 1 - probchi( upp, df );
      arl_f = 1 / p;
      output;
   end;
run;
```

The data set ARLSHIFT is listed in Output 47.2.2. For example, on average, 53 samples are required to detect a ten percent increase in σ (a shifted standard deviation of approximately 219). The computations use the fact that $(n_i - 1)s_i^2/\sigma^2$ has a χ^2 distribution with $n_i - 1$ degrees of freedom, assuming that the measurements are normally distributed.

Output 47.2.2. The Data Set ARLSHIFT

Average Run Length Analysis

f	f_std	p	arl_f
1.00	198.996	0.00279	358.073
1.05	208.945	0.00758	131.922
1.10	218.895	0.01875	53.322
1.15	228.845	0.03984	25.102
1.20	238.795	0.07388	13.535
1.25	248.745	0.12239	8.171
1.30	258.694	0.18475	5.413
1.35	268.644	0.25834	3.871
1.40	278.594	0.33923	2.948
1.45	288.544	0.42298	2.364
1.50	298.494	0.50546	1.978

Chapter 48
UCHART Statement

Chapter Contents

OVERVIEW 1651

GETTING STARTED 1652
- Creating u Charts from Defect Count Data 1652
- Saving Control Limits 1654
- Reading Preestablished Control Limits 1656
- Creating u Charts from Nonconformities per Unit 1657
- Saving Nonconformities per Unit 1660

SYNTAX 1662
- Summary of Options 1664

DETAILS 1673
- Constructing Charts for Nonconformities per Unit (u Charts) 1673
- Output Data Sets 1675
- ODS Tables 1678
- Input Data Sets 1678
- Axis Labels 1681
- Missing Values 1681

EXAMPLES 1682
- Example 48.1. Applying Tests for Special Causes 1682
- Example 48.2. Specifying a Known Expected Number of Nonconformities . 1684
- Example 48.3. Creating u Charts for Varying Numbers of Units 1685

Chapter 48
UCHART Statement

Overview

The UCHART statement creates u charts for the numbers of nonconformities (defects) per inspection unit in subgroup samples containing arbitrary numbers of units.

You can use options in the UCHART statement to

- specify the number of inspection units per subgroup
- compute control limits from the data based on a multiple of the standard error of the plotted values or as probability limits
- tabulate subgroup summary statistics and control limits
- save control limits in an output data set
- save subgroup summary statistics in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify a known (standard) value for the average number of nonconformities per inspection unit
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

Getting Started

This section introduces the UCHART statement with simple examples that illustrate commonly used options. Complete syntax for the UCHART statement is presented in the "Syntax" section on page 1662, and advanced examples are given in the "Examples" section on page 1682.

Creating u Charts from Defect Count Data

See SHWUCHR in the SAS/QC Sample Library

A textile company uses a u chart to monitor the number of defects per square meter of fabric. The fabric is spooled onto rolls as it is inspected for defects. Each piece of fabric is one meter wide and 30 meters in length. The following statements create a SAS data set named FABRIC, which contains the defect counts for 20 rolls:

```
data fabric;
   input roll defects @@;
datalines;
 1 12    2 11    3  9    4 15
 5  7    6  6    7  5    8 10
 9  8   10  8   11 14   12  5
13  9   14 13   15  7   16  5
17  8   18 11   19  7   20 12
;
run;
```

A partial listing of FABRIC is shown in Figure 48.1.

Number of Fabric Defects

roll	defects
1	12
2	11
3	9
4	15
5	7
.	.
.	.
.	.

Figure 48.1. The Data Set FABRIC

There is a single observation per roll. The variable ROLL identifies the subgroup sample and is referred to as the *subgroup-variable*. The variable DEFECTS contains the number of nonconformities (defect count) for each subgroup sample and is referred to as the *process variable* (or *process* for short).

The following statements create the u chart shown in Figure 48.2:

```
title 'u Chart for Fabric Defects';
proc shewhart data=fabric;
   uchart defects*roll / subgroupn = 30;
run;
```

This example illustrates the basic form of the UCHART statement. After the keyword UCHART, you specify the *process* to analyze (in this case, DEFECTS), followed by an asterisk and the *subgroup-variable* (ROLL).

The SUBGROUPN= option specifies the number of inspection units in each subgroup sample and is required if the input data set is a DATA= data set. In this example, each square meter of fabric is an inspection unit, and each roll is a subgroup sample. The number of inspection units per subgroup can be thought of as the subgroup sample size.

You can use the SUBGROUPN= option to specify one of the following:

- a constant subgroup sample size (as in this example)
- an input variable name whose values contain the subgroup sample sizes (for an example, see "Saving Nonconformities per Unit" on page 1660)

Options such as SUBGROUPN= are specified after the slash (/) in the UCHART statement. A complete list of options is presented in the "Syntax" section on page 1662.

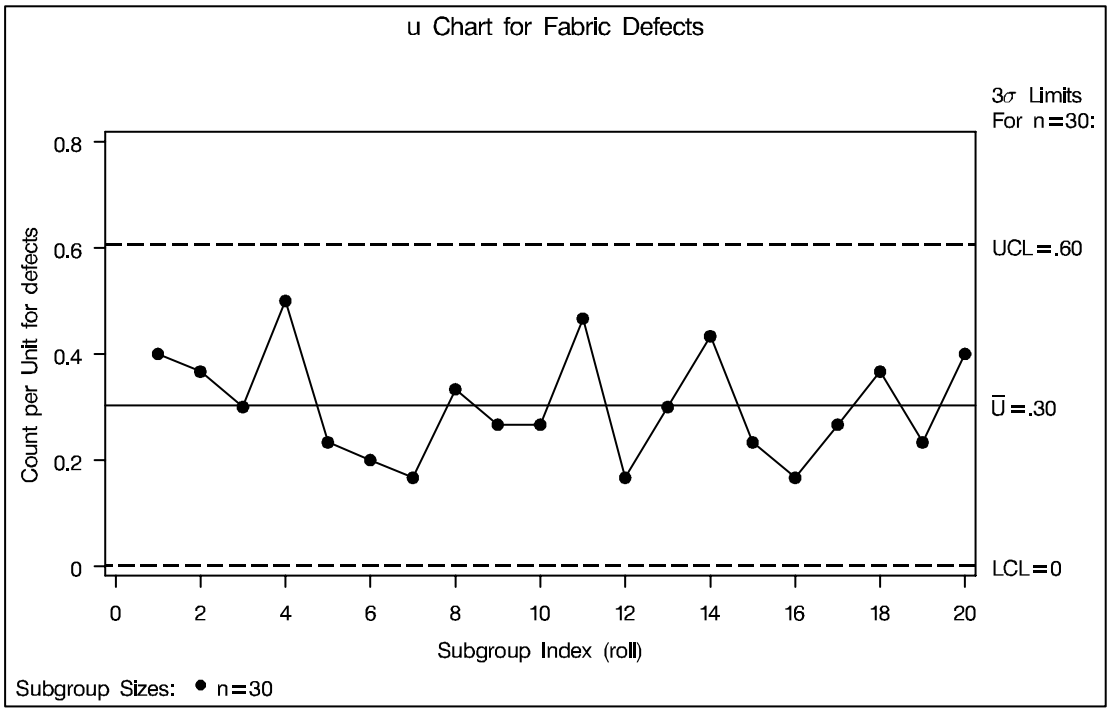


Figure 48.2. u Chart Example

The input data set is specified with the DATA= option in the PROC SHEWHART statement.

Each point on the u chart represents the number of nonconformities per inspection unit for a particular subgroup. For instance, the value plotted for the first subgroup is $12/30 = 0.4$ (since there are 12 defects on the first roll and this roll contains 30

square meters of fabric). By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given on page 1674. Since none of the points exceed the 3σ limits, the u chart indicates that the fabric manufacturing process is in statistical control.

See "Constructing Charts for Nonconformities per Unit (u Charts)" on page 1673 for details concerning u charts. For more details on reading defect count data, see "DATA= Data Set" on page 1678.

Saving Control Limits

See SHWUCHR in the SAS/QC Sample Library

You can save the control limits for a u chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1656) or modify the limits with a DATA step program.

The following statements read defect counts from the data set FABRIC (see page 1652) and save the control limits displayed in Figure 48.2 in a data set named FABLIM:

```
proc shewhart data=fabric;
   uchart defects*roll / subgroupn = 30
                         outlimits = fablim
                         nochart;
run;
```

The SUBGROUPN= option specifies the number of inspection units in each subgroup sample. The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. The data set FABLIM is listed in Figure 48.3.

Control Limits Data Set FABLIM

VAR	_SUBGRP_	_LIMITN_	_SIGMAS_	_LCLU_	_U_	_UCLU_	_LCLC_	_UCLC_
defects	roll	ESTIMATE	30	.002550178	3	.001671271	0.30333	0.60500

Figure 48.3. The Data Set FABLIM Containing Control Limit Information

The data set FABLIM contains one observation with the limits for *process* DEFECTS. The variables _LCLU_ and _UCLU_ contain the lower and upper control limits, and the variable _U_ contains the central line. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*.

The variable _TYPE_ is a bookkeeping variable that indicates whether the value of _U_ is an estimate or standard value. For more information, see "OUTLIMITS= Data Set" on page 1675.

Alternatively, you can use the OUTTABLE= option to create an output data set that saves both the control limits and the subgroup statistics, as illustrated by the following statements:

```
proc shewhart data=fabric;
   uchart defects*roll / subgroupn = 30
                         outtable  = fabtab
                         nochart;
run;
```

The data set FABTAB is listed in Figure 48.4.

Number of Defects Per Square Meter and Control Limits

VAR	roll	_SIGMAS_	_SUBN_	_SUBU_	_LCLU_	_SUBU_	_U_	_UCLU_	_EXLIM_
defects	1	3	30	30	.001671271	0.40000	0.30333	0.60500	
defects	2	3	30	30	.001671271	0.36667	0.30333	0.60500	
defects	3	3	30	30	.001671271	0.30000	0.30333	0.60500	
defects	4	3	30	30	.001671271	0.50000	0.30333	0.60500	
defects	5	3	30	30	.001671271	0.23333	0.30333	0.60500	
defects	6	3	30	30	.001671271	0.20000	0.30333	0.60500	
defects	7	3	30	30	.001671271	0.16667	0.30333	0.60500	
defects	8	3	30	30	.001671271	0.33333	0.30333	0.60500	
defects	9	3	30	30	.001671271	0.26667	0.30333	0.60500	
defects	10	3	30	30	.001671271	0.26667	0.30333	0.60500	
defects	11	3	30	30	.001671271	0.46667	0.30333	0.60500	
defects	12	3	30	30	.001671271	0.16667	0.30333	0.60500	
defects	13	3	30	30	.001671271	0.30000	0.30333	0.60500	
defects	14	3	30	30	.001671271	0.43333	0.30333	0.60500	
defects	15	3	30	30	.001671271	0.23333	0.30333	0.60500	
defects	16	3	30	30	.001671271	0.16667	0.30333	0.60500	
defects	17	3	30	30	.001671271	0.26667	0.30333	0.60500	
defects	18	3	30	30	.001671271	0.36667	0.30333	0.60500	
defects	19	3	30	30	.001671271	0.23333	0.30333	0.60500	
defects	20	3	30	30	.001671271	0.40000	0.30333	0.60500	

Figure 48.4. The Data Set FABTAB

This data set contains one observation for each subgroup sample. The variables _SUBU_ and _SUBN_ contain the number of nonconformities per unit in each subgroup and the number of inspection units per subgroup. The variables _LCLU_ and _UCLU_ contain the lower and upper control limits, and the variable _U_ contains the central line. The variables _VAR_ and ROLL contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1677.

An OUTTABLE= data set can be read later as a TABLE= data set by the SHEWHART procedure. For example, the following statements read FABTAB and display a u chart (not shown here) identical to the chart in Figure 48.2:

```
title 'u Chart for Fabric Defects';
proc shewhart table=fabtab;
   uchart defects*roll / subgroupn=30;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see Chapter 56, "Specialized Control Charts,"). For more information, see "TABLE= Data Set" on page 1680.

Reading Preestablished Control Limits

See SHWUCHR in the SAS/QC Sample Library

In the previous example, control limits were saved in a SAS data set named FABLIM. This example shows how these limits can be applied to defect counts for an additional 20 rolls of fabric, which are provided in the following data set:

```
data fabric2;
   input roll defects @@;
   datalines;
21  9    22  9    23 12    24  7    25 14
26  5    27  4    28 13    29  6    30  7
31 11    32 10    33  7    34 11    35  8
36  8    37  3    38 10    39  9    40 13
;
run;
```

The following statements create a u chart for the second group of rolls using the control limits in FABLIM:

```
title 'u Chart for Fabric Defects';
proc shewhart data=fabric2 limits=fablim;
   uchart defects*roll / subgroupn = 30;
run;
```

The chart is shown in Figure 48.5 and indicates that the process is in control.

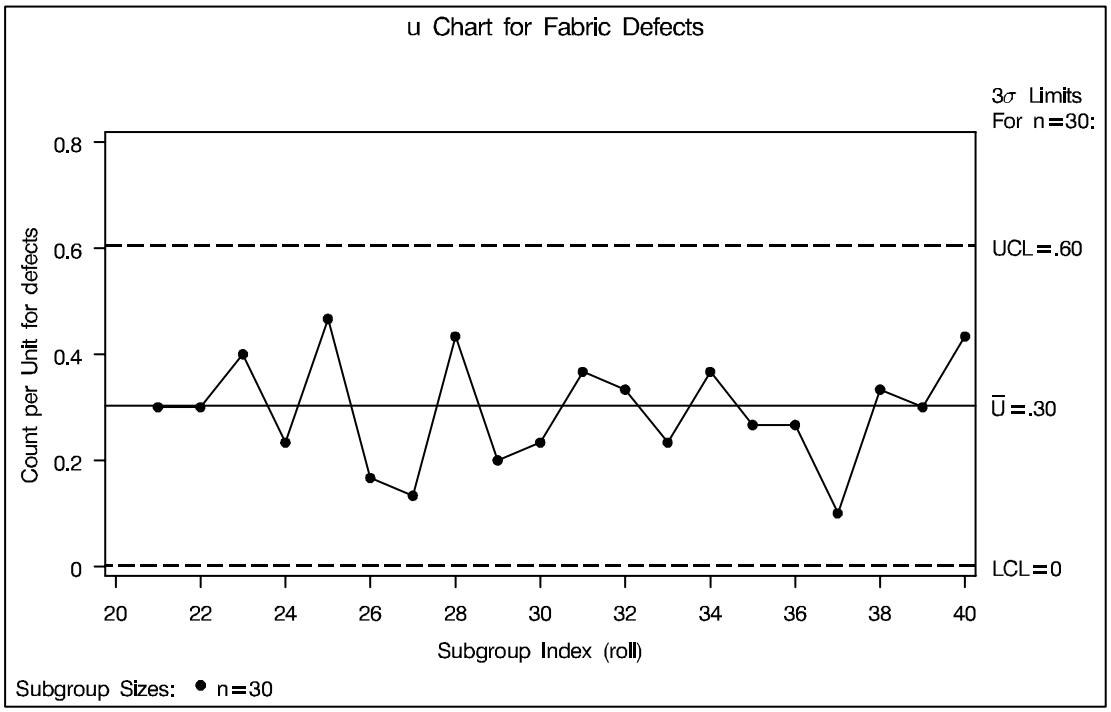


Figure 48.5. A u Chart for Second Set of Fabric Rolls

The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default,* this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* DEFECTS
- the value of _SUBGRP_ matches the *subgroup-variable* name ROLL

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1679 for details concerning the variables that you must provide.

Creating u Charts from Nonconformities per Unit

See SHWUCHR in the SAS/QC Sample Library

In the previous example, the input data set provided the number of nonconformities for each subgroup sample. However, in some applications, as illustrated here, the data provide the number of nonconformities *per inspection unit* for each subgroup.

A clothing manufacturer ships shirts in boxes of ten. Prior to shipment, each shirt is inspected for flaws. Since the manufacturer is interested in the average number of flaws per shirt, the number of flaws found in each box is divided by ten and then

*In Release 6.09 and in earlier releases, it is also necessary to specify the READLIMITS option to read control limits from a LIMITS= data set.

recorded. The following statements create a SAS data set named SHIRTS, which contains the average number of flaws per shirt for 25 boxes:

```
data shirts;
   input box avgdefu @@;
   avgdefn=10;
   datalines;
 1  0.4    2  0.7    3  0.5    4  1.0    5  0.3
 6  0.2    7  0.0    8  0.4    9  0.4   10  0.6
11  0.2   12  0.7   13  0.3   14  0.1   15  0.3
16  0.6   17  0.6   18  0.3   19  0.7   20  0.3
21  0.0   22  0.1   23  0.5   24  0.6   25  0.4
;
run;
```

Note that this is the same data set used in "Getting Started" of Chapter 40, "CCHART Statement." A partial listing of SHIRTS is shown in Figure 48.6.

Average Number of Shirt Flaws

box	avgdefu	avgdefn
1	0.4	10
2	0.7	10
3	0.5	10
.	.	.
.	.	.
.	.	.

Figure 48.6. The Data Set SHIRTS

The data set SHIRTS contains three variables: the box number (BOX), the average number of flaws per shirt (AVGDEFU), and the number of shirts per box (AVGDEFN). Here, a *subgroup* is a box of shirts, and an *inspection unit* is an individual shirt. Note that each subgroup contains ten inspection units.

To create a u chart for the average number of flaws per shirt in each box, you can specify SHIRTS as a HISTORY= data set.

```
title 'Total Flaws per Box of Shirts';
proc shewhart history=shirts;
   uchart avgdef*box ;
run;
```

Note that AVGDEF is *not* the name of a SAS variable in the data set but is, instead, the common prefix for the names of the SAS variables AVGDEFU and AVGDEFN. The suffix characters *U* and *N* indicate *number of nonconformities per unit* and *sample size*, respectively. This naming convention enables you to specify two variables in the HISTORY= data set with a single name, which is referred to as the *process*. The name BOX, specified after the asterisk, is the name of the *subgroup-variable*. The u chart is shown in Figure 48.7.

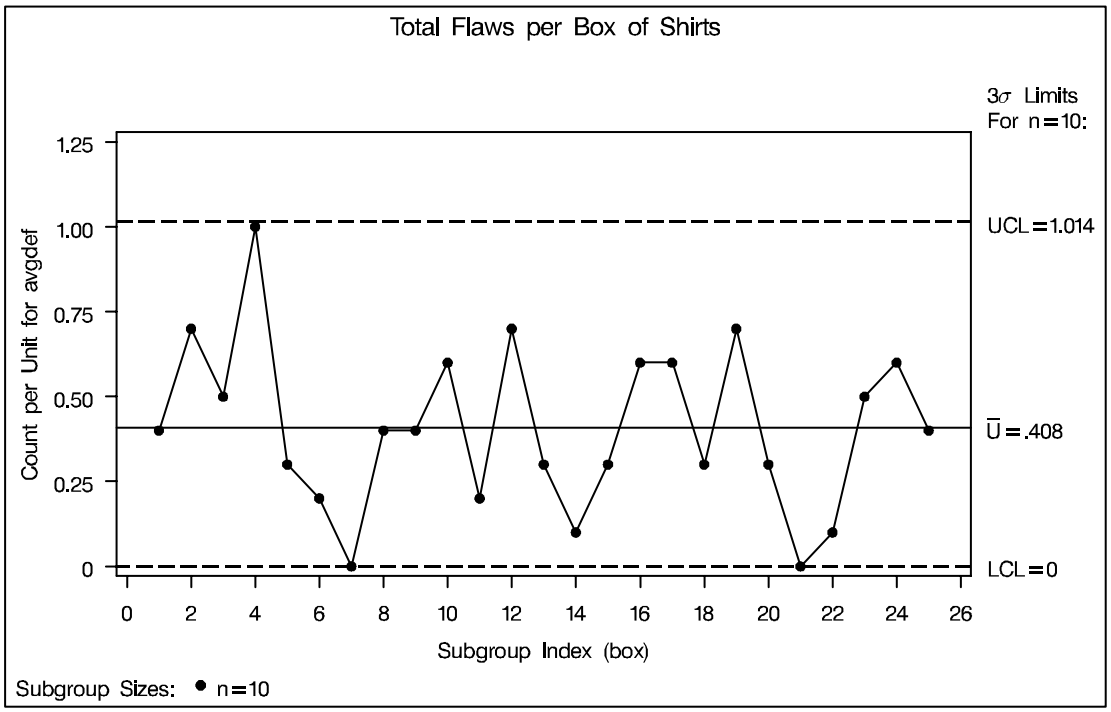


Figure 48.7. A u Chart for Boxes of Shirts

In general, a HISTORY= input data set used with the UCHART statement must contain the following variables:

- subgroup variable
- subgroup number of nonconformities per unit variable
- subgroup sample size variable

Furthermore, the names of the nonconformities per unit and sample size variables must begin with the *process* name specified in the UCHART statement and end with the special suffix characters *U* and *N*, respectively. If the names do not follow this convention, you can use the RENAME option to rename the variables for the duration of the SHEWHART procedure step. Suppose that, instead of the variables AVGDEFU and AVGDEFN, the data set SHIRTS contained the variables SHIRTDEF and SIZES. The following statements temporarily rename SHIRTDEF and SIZES to AVGDEFU and AVGDEFN:

```
proc shewhart
   history=shirts (rename=(shirtdef = avgdefu
                           sizes    = avgdefn ));
   uchart avgdef*box;
run;
```

For more information, see "HISTORY= Data Set" on page 1679.

Saving Nonconformities per Unit

See SHWUCHR in the SAS/QC Sample Library

In this example, the UCHART statement is used to create a summary data set containing the number of nonconformities per unit. This data set can be read later by the SHEWHART procedure (as in the preceding example).

A department store receives boxes of shirts containing 10, 25, or 50 shirts. Each box is inspected, and the total number of defects per box is recorded. The following statements create a SAS data set named SHIRTS2, which contains the total defects per box for 20 boxes:

```
data shirts2;
   input box flaws nshirts @@;
   datalines;
 1  3  10    2  8  10    3 15  25    4 20  25
 5  9  25    6  1  10    7  1  10    8 21  50
 9  3  10   10  7  10   11  1  10   12 21  25
13  9  25   14  3  25   15 12  50   16 18  50
17  7  10   18  4  10   19  8  10   20  4  10
;
run;
```

A partial listing of SHIRTS2 is shown in Figure 48.8.

Number of Shirt Flaws per Box

box	avgdefu	avgdefn
1	0.4	10
2	0.7	10
3	0.5	10
4	1.0	10
5	0.3	10
.	.	.
.	.	.
.	.	.

Figure 48.8. The Data Set SHIRTS2

The variable BOX contains the box number, the variable FLAWS contains the number of flaws in each box, and the variable NSHIRTS contains the number of shirts in each box. To evaluate the quality of the shirts, you should report the average number of defects per shirt. The following statements create a data set containing the number of flaws per shirt and the number of shirts per box:

```
proc shewhart data=shirts2;
   uchart flaws*box / subgroupn  = nshirts
                      outhistory = shrthist
                      nochart;
run;
```

The SUBGROUPN= option names the variable in the DATA= data set whose values specify the number of inspection units per subgroup. The OUTHISTORY= option names an output data set containing the number of nonconformities per inspection unit and the number of inspection units per subgroup. A partial listing of SHRTHIST is shown in Figure 48.9.

Average Defects Per Tee Shirt

box	flaws U	flaws N
1	0.30	10
2	0.80	10
3	0.60	25
4	0.80	25
5	0.36	25
.	.	.
.	.	.
.	.	.

Figure 48.9. The Data Set SHRTHIST

There are three variables in the data set SHRTHIST.

- BOX contains the subgroup index.
- FLAWSU contains the numbers of nonconformities per inspection unit.
- FLAWSN contains the subgroup sample sizes.

Note that the variables containing the numbers of nonconformities per inspection unit and subgroup sample sizes are named by adding the suffix characters *U* and *N* to the *process* FLAWS specified in the UCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 1676.

Syntax

The basic syntax for the UCHART statement is as follows:

UCHART *process******subgroup-variable* **;**

The general form of this syntax is as follows:

UCHART (*processes***)***subgroup-variable* <**(***block-variables* **)** >
< =*symbol-variable* | =*'character'* > < **/** *options* >**;**

You can use any number of UCHART statements in the SHEWHART procedure. The components of the UCHART statement are described as follows.

process
processes
identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If numbers of nonconformities per subgroup are read from a DATA= data set, *process* must be the name of the variable containing the numbers of nonconformities. For an example, see "Creating u Charts from Defect Count Data" on page 1652.
- If numbers of nonconformities per unit and numbers of inspection units per subgroup are read from a HISTORY= data set, *process* must be the common prefix of the appropriate variables in the HISTORY= data set. For an example, see "Creating u Charts from Nonconformities per Unit" on page 1657.
- If numbers of nonconformities per item, numbers of inspection units per subgroup, and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1654.

A *process* is required. If you specify more than one process, enclose the list in parentheses. For example, the following statements request distinct u charts for DEFECTS and FLAWS:

```
proc shewhart data=measures;
   uchart (defects flaws)*sample / subgroupn=50;
run;
```

Note that when data are read from a DATA= data set with the UCHART statement, the SUBGROUPN= option (which specifies the number of inspection units per subgroup) is required.

subgroup-variable
is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding UCHART statement, SAMPLE is the subgroup variable. For details, see "Subgroup Variables" on page 1771.

block-variables
are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 1932 for an example.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the number of nonconformities per unit.

- If you produce a chart on a line printer, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce a chart on a graphics device, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOLn statements. See "Displaying Stratification in Levels of a Classification Variable" on page 1931 for an example.

character
specifies a plotting character for charts produced on line printers. For example, the following statements create a u chart using an asterisk (*) to plot the points:

```
proc shewhart data=values;
   uchart defects*sample='*' / subgroupn=100;
run;
```

options
enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The "Summary of Options" section, which follows, lists all options by function. Chapter 53, "Dictionary of Options," describes each option in detail.

Summary of Options

The following tables list the UCHART statement options by function. For complete descriptions, see Chapter 53, "Dictionary of Options."

Table 48.1. Tabulation Options

TABLE	creates a basic table of subgroup sample sizes, subgroup numbers of nonconformities per unit, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with a column indicating which tests for special causes are positive

Note that specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.

Table 48.2. Options for Specifying Tests for Special Causes

NO3SIGMACHECK	allows tests to be applied with control limits other than 3σ limits
TESTS=*value-list*|*customized-pattern-list*	specifies tests for special causes
TEST2RUN=*n*	specifies length of pattern for Test 2
TEST3RUN=*n*	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL='*label*'|(*variable*)|*keyword*	provides labels for points where test is positive
TESTLABEL*n*='*label*'	specifies label for n^{th} test for special causes
TESTNMETHOD=STANDARDIZE	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
TESTRESET=*variable*	allows tests for special causes to be reset
ZONELABELS	adds labels A, B, and C to zone lines
ZONES	adds lines delineating zones A, B, and C
ZONEVALPOS=*n*	specifies position of ZONEVALUES labels
ZONEVALUES	labels zone lines with their values

Table 48.3. Graphical Options for Displaying Tests for Special Causes

CTESTLABBOX=*color*	specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=*color*| *test-color-list*	specifies color for labels indicating points where test is positive
CTESTSYMBOL=*color*	specifies color for symbol used to plot points where test is positive
CZONES=*color*	specifies color for lines and labels delineating zones A, B, and C
LTESTS=*linetype*	specifies type of line connecting points where test is positive
LZONES=*linetype*	specifies line type for lines delineating zones A, B, and C
TESTFONT=*font*	specifies software font for labels at points where test is positive
TESTHEIGHT=*value*	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=*symbol*	specifies plot symbol for points where test is positive
TESTSYMBOLHT=*value*	specifies symbol height for points where test is positive
WTESTS=*n*	specifies width of line connecting points where test is positive

Table 48.4. Line Printer Options for Displaying Tests for Special Causes

TESTCHAR='*character*'	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
ZONECHAR='*character*'	specifies character for lines that delineate zones for tests for special causes

Table 48.5. Reference Line Options

CHREF=*color*	specifies color for lines requested by the HREF= option
CVREF=*color*	specifies color for lines requested by the VREF= option
HREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on u chart
HREFCHAR='*character*'	specifies line character for HREF= lines
HREFDATA= *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on u chart
HREFLABELS= *'label1'...'labeln'*	specifies labels for HREF= lines
HREFLABPOS=*n*	specifies position of HREFLABELS= labels
LHREF=*linetype*	specifies line type for HREF= lines
LVREF=*linetype*	specifies line type for VREF= lines
NOBYREF	specifies that reference line information in a data set is to be applied uniformly to charts created for all BY groups
VREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to vertical axis
VREFCHAR='*character*'	specifies line character for VREF= lines
VREFLABELS= *'label1'...'labeln'*	specifies labels for VREF= lines
VREFLABPOS=*n*	specifies position of VREFLABELS= labels

Table 48.6. Clipping Options

CCLIP=*color*	specifies color for plot symbol for clipped points
CLIPCHAR='*character*'	specifies plot character for clipped points
CLIPFACTOR=*value*	determines extent to which extreme points are clipped
CLIPLEGEND='*string*'	specifies text for clipping legend
CLIPLEGPOS=*keyword*	specifies position of clipping legend
CLIPSUBCHAR= '*character*'	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=*symbol*	specifies plot symbol for clipped points
CLIPSYMBOLHT=*value*	specifies symbol marker height for clipped points

Table 48.7. Axis and Axis Label Options

CAXIS=*color*	specifies color for axis lines and tick marks
CFRAME=*color*|(*color-list*)	specifies fill colors for frame for plot area
CTEXT=*color*	specifies color for tick mark values and axis labels
HAXIS=*values*|AXIS*n*	specifies major tick mark values for horizontal axis
HEIGHT=*value*	specifies height of axis label and axis legend text
HMINOR=*n*	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=*value*	specifies length of offset at both ends of horizontal axis
INTSTART=*value*	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	suppresses vertical axis truncation at zero applied by default
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=*n*	specifies thinning factor for tick mark labels on horizontal axis
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=*values*|AXIS*n*	specifies major tick mark values for vertical axis
VFORMAT=*format*	specifies format for vertical axis tick mark labels
VMINOR=*n*	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=*value*	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis
WAXIS=*n*	specifies width of axis lines

Table 48.8. Options for Specifying Control Limits

ALPHA=*value*	requests probability limits for control charts
LIMITN=*n*|VARYING	specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	computes control limits for each *process* from the data rather than from a LIMITS= data set (Release 6.10 and later releases)
READALPHA	reads the variable _ALPHA_ instead of the variable _SIGMAS_ from a LIMITS= data set
READINDEXES=ALL| '*label1*'...'*labeln*'	reads multiple sets of control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (Release 6.09 and earlier releases)
SIGMAS=*k*	specifies width of control limits in terms of multiple k of standard error of plotted statistic

Table 48.9. Options for Displaying Control Limits

CINFILL=*color*	specifies color for area inside control limits
CLIMITS=*color*	specifies color of control limits, central line, and related labels
LCLLABEL='*label*'	specifies label for lower control limit
LIMLABSUBCHAR= '*character*'	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=*linetype*	specifies line type for control limits
NDECIMAL=*n*	specifies number of digits to right of decimal place in default labels for control limits and central line
NOCTL	suppresses display of central line on u chart
NOLCL	suppresses display of lower control limit
NOLIMITLABEL	suppresses labels for control limits and central line
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOLIMIT0	suppresses display of zero lower control limit for u chart
NOUCL	suppresses display of upper control limit
UCLLABEL='*string*'	specifies label for upper control limit
USYMBOL='*string*'| *keyword*	specifies label for central line
WLIMITS=*n*	specifies width for control limits and central line

Table 48.10. Grid Options

CGRID=*color*	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=*linetype*	specifies line type for grid requested with the ENDGRID option
LGRID=*linetype*	specifies line type for grid requested with the GRID option
WGRID=*n*	specifies width of grid lines

Table 48.11. Options for Plotting and Labeling Points

ALLLABEL=VALUE|(*variable*)	labels every point
CLABEL=*color*	specifies color for labels
CCONNECT=*color*	specifies color for line segments that connect points on chart
CFRAMELAB=*color*	specifies fill color for frame around labeled points
CNEEDLES=*color*	specifies color for needles that connect points to central line
CONNECTCHAR='*character*'	specifies character used to form line segments that connect points on chart
COUT=*color*	specifies color for portions of line segments that connect points outside control limits
COUTFILL=*color*	specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=*angle*	specifies angle at which labels are drawn
LABELFONT=*font*	specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=*value*	specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on chart
OUTLABEL=VALUE|(*variable*)	labels points outside control limits
SYMBOLCHARS='*characters*'	specifies characters indicating *symbol-variable*
SYMBOLLEGEND=NONE|*name*	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=*keyword*	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=*n*	specifies width of needles

Table 48.12. Options for Interactive Control Charts

HTML=(*variable*)	specifies a variable whose values are URLs to be associated with subgroups
HTML_LEGEND=(*variable*)	specifies a variable whose values are URLs to be associated with symbols in the symbol legend
WEBOUT=*SAS-data-set*	creates an OUTTABLE= data set with additional graphics coordinate data

Table 48.13. Block Variable Legend Options

BLOCKLABELPOS=*keyword*	specifies position of label for *block-variable* legend
BLOCKLABTYPE=*value*|*keyword*	specifies text size of *block-variable* legend
BLOCKPOS=*n*	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=*color-list*	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=*variable*|(*variables*)	specifies one or more variables whose values are colors for filling background of *block-variable* legend

Table 48.14. Phase Options

CPHASELEG=*color*	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE='*string*'	specifies value of the variable _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=*value*|*keyword*	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES= ALL|'*label1*'...'*labeln*'	specifies *phases* to be read from an input data set

Table 48.15. Standard Value Options

TYPE=*keyword*	identifies whether parameters are estimates or standard values and specifies value of the variable _TYPE_ in the OUTLIMITS= data set
U0=*value*	specifies known average number of nonconformities per unit

Table 48.16. Input Data Set Option

MISSBREAK	specifies that observations with missing values are not to be processed
SUBGROUPN=*n*| *variable*	specifies subgroup sample sizes as constant number n or as values of *variable* in the DATA= data set

Table 48.17. Output Data Set Options

OUTHISTORY= *SAS-data-set*	creates output data set containing subgroup numbers of nonconformities per unit and subgroup sample sizes
OUTINDEX='*string*'	specifies value of the variable _INDEX_ in the OUTLIMITS= data set
OUTLIMITS= *SAS-data-set*	creates output data set containing control limits
OUTTABLE= *SAS-data-set*	creates output data set containing subgroup numbers of nonconformities per unit, subgroup sample sizes, and control limits

Table 48.18. Plot Layout Options

ALLN	plots numbers of nonconformities per unit for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts for a process variable only when exceptions occur
INTERVAL=*keyword*	specifies natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=*n*	specifies maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=*n*	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=*n*	specifies number of pages or screens to be used to display chart
ZEROSTD	displays u chart regardless of whether $\hat{\sigma} = 0$

Table 48.19. Graphical Enhancement Options

ANNOTATE=*SAS-data-set*	specifies annotate data set that adds features to chart
DESCRIPTION='*string*'	specifies string that appears in the description field of the PROC GREPLAY master menu
FONT=*font*	specifies software font for labels and legends on charts
NAME='*string*'	specifies name that appears in the name field of the PROC GREPLAY master menu
PAGENUM='*string*'	specifies the form of the label used in pagination
PAGENUMPOS=*keyword*	specifies the position of the page number requested with the PAGENUM= option

Table 48.20. Star Options

CSTARCIRCLES=*color*	specifies color for circles specified by the STARCIRCLES= option
CSTARFILL=*color*|*(variable)*	specifies color for filling stars
CSTAROUT=*color*	specifies outline color for stars exceeding inner or outer circles
CSTARS=*color*| *(variable)*	specifies color for outlines of stars
LSTARCIRCLES=*linetypes*	specifies line types STARCIRCLES= circles
LSTARS=*linetype*|*(variable)*	specifies line types for outlines of stars requested with the STARVERTICES= option
STARBDRADIUS=*value*	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=*value-list*	specifies reference circles for stars
STARINRADIUS=*value*	specifies inner radius of stars
STARLABEL=*keyword*	specifies vertices to be labeled
STARLEGEND=*keyword*	specifies style of legend for star vertices
STARLEGENDLAB='*label*'	specifies label for STARLEGEND= legend
STAROUTRADIUS=*value*	specifies outer radius of stars
STARSPEC=*value*|*SAS-data-set*	specifies method used to standardize vertex variables
STARSTART=*value*	specifies angle for first vertex
STARTYPE=*keyword*	specifies graphical style of star
STARVERTICES=*variable*|*(variables)*	superimposes star at each point on chart
WSTARCIRCLES=*n*	specifies width of circles requested by the STARCIRCLES= option
WSTARS=*n*	specifies width of stars requested by the STARVERTICES= option

Table 48.21. Overlay Options

CCOVERLAY=*color-list*	specifies colors for overlay line segments
COVERLAY=*color-list*	specifies colors for overlay plots
COVERLAYCLIP=*color*	specifies color for clipped points on overlays
LOVERLAY=*linetypes*	specifies line types for overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=*variable-list*	specifies variables to overlay on control chart
OVERLAYCLIPSYM=*symbol*	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=*value*	specifies symbol height for clipped points on overlays
OVERLAYHTML=*variable-list*	specifies URLS to associate with overlay points
OVERLAYID=*variable-list*	specifies labels for primary chart overlay points
OVERLAYLEGLAB='*label*'	specifies label for overlay legend
OVERLAYSYM=*symbol-list*	specifies symbols for overlay plots
OVERLAYSYMHT=*value-list*	specifies symbol heights for overlay plots
WOVERLAY=*value-list*	specifies widths of overlay line segments

Details

Constructing Charts for Nonconformities per Unit (u Charts)

The following notation is used in this section:

u	expected number of nonconformities per unit produced by process
u_i	number of nonconformities per unit in the i^{th} subgroup. In general, $u_i = c_i/n_i$.
c_i	total number of nonconformities in the i^{th} subgroup
n_i	number of inspection units in the i^{th} subgroup
$\bar{u}$	average number of nonconformities per unit taken across subgroups. The quantity $\bar{u}$ is computed as a weighted average: $$\bar{u} = \frac{n_1 u_1 + \cdots + n_N u_N}{n_1 + \cdots + n_N} = \frac{c_1 + \cdots + c_N}{n_1 + \cdots + n_N}$$
N	number of subgroups
χ^2_ν	has a central χ^2 distribution with ν degrees of freedom

Plotted Points

Each point on a u chart indicates the number of nonconformities per unit (u_i) in a subgroup. For example, Figure 48.10 displays three sections of pipeline that are inspected for defective welds (indicated by an *X*). Each section represents a *subgroup* composed of a number of *inspection units*, which are 1000-foot-long sections. The number of units in the i^{th} subgroup is denoted by n_i, which is the subgroup sample size.

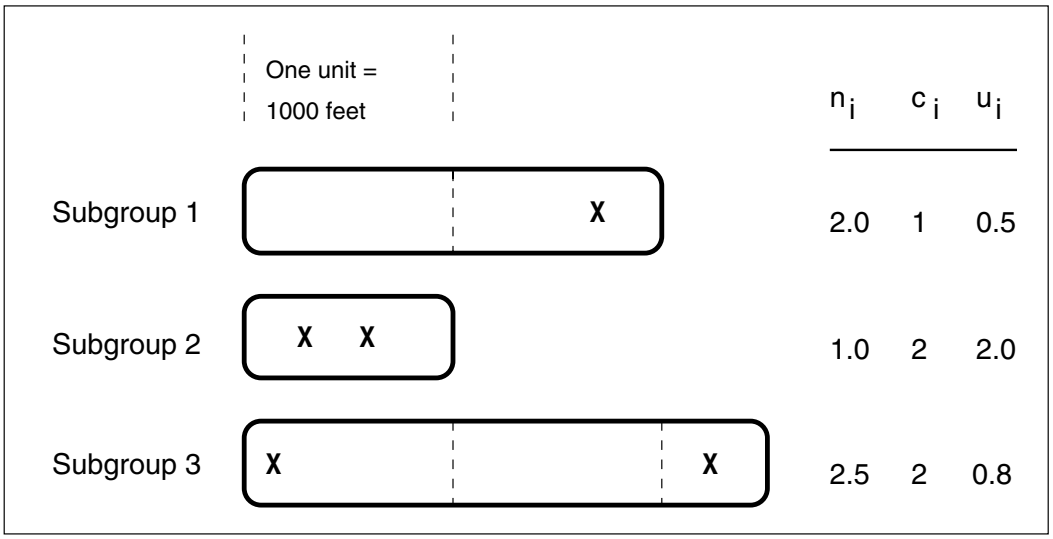


Figure 48.10. Terminology for c Charts and u Charts

The *number of nonconformities* in the i^{th} subgroup is denoted by c_i. The *number of nonconformities per unit* in the i^{th} subgroup is denoted by $u_i = c_i/n_i$. In Figure 48.10, the number of defective welds per unit in the third subgroup is $u_3 = 2/2.5 = 0.8$.

A u chart plots the quantity u_i for the i^{th} subgroup. A c chart plots the quantity c_i for the i^{th} subgroup (see Chapter 40, "CCHART Statement,"). An advantage of a u chart is that the value of the central line at the i^{th} subgroup does not depend on n_i. This is not the case for a c chart, and consequently, a u chart is often preferred when the number of units n_i is not constant across subgroups.

Central Line

On a u chart, the central line indicates an estimate of u, which is computed as $\bar{u}$ by default. If you specify a known value (u_0) for u, the central line indicates the value of u_0.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of u_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that u_i exceeds the limits

The lower and upper control limits, LCLU and UCLU, respectively, are given by

$$\begin{aligned} \text{LCLU} &= \max\left(\bar{u} - k\sqrt{\bar{u}/n_i}\,, 0\right) \\ \text{UCLU} &= \bar{u} + k\sqrt{\bar{u}/n_i} \end{aligned}$$

The limits vary with n_i.

The upper probability limit UCLU for u_i can be determined using the fact that

$$\begin{aligned} P\{u_i > \text{UCLU}\} &= 1 - P\{u_i \leq \text{UCLU}\} \\ &= 1 - P\{c_i \leq n_i\text{UCLU}\} \\ &= 1 - P\{\chi^2_{2(n_i(\text{UCLU}+1))} \geq 2n_i\bar{u}\} \end{aligned}$$

The limit UCLU is then calculated by setting

$$1 - P\{\chi^2_{2(n_i(\text{UCLU}+1))} \geq 2n_i\bar{u}\} = \alpha/2$$

and solving for UCLU.

Likewise, the lower probability limit LCLC for u_i can be determined using the fact that

$$\begin{aligned} P\{u_i < \text{LCLC}\} &= P\{c_i < n_i\text{LCLU}\} \\ &= P\{\chi^2_{2(n_i(\text{LCLC}+1)} > 2n_i\bar{u}\} \end{aligned}$$

The limit LCLC is then calculated by setting

$$P\{\chi^2_{2(n_i(\text{LCLC}+1)} > 2n_i\bar{u}\} = \alpha/2$$

and solving for LCLC. For more information, refer to Johnson, Kotz, and Kemp (1992). This assumes that the process is in statistical control and that c_i has a Poisson distribution. Note that the probability limits vary with n_i and are asymmetric around the central line. If a standard value u_0 is available for u, replace $\bar{u}$ with u_0 in the formulas for the control limits.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify u_0 with the U0= option or with the variable _U_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables can be saved:

Table 48.22. OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLU	lower control limit for number of nonconformities per unit
LIMITN	sample size associated with the control limits
SIGMAS	multiple (k) of standard error of u_i
SUBGRP	*subgroup-variable* specified in the UCHART statement
TYPE	type (estimate or standard value) of _U_
_U_	value of central line of u chart ($\bar{u}$ or u_0)
UCLU	upper control limit for number of nonconformities per unit
VAR	*process* specified in the UCHART statement

Notes:

1. If the control limits vary with subgroup sample size, the special missing value *V* is assigned to the variables _LCLU_, _UCLU_, and _LIMITN_.
2. If the limits are defined in terms of a multiple k of the standard error of u_i, the value of _ALPHA_ is computed as $P\{u_i < \text{\_LCLU\_}\} + P\{u_i > \text{\_UCLU\_}\}$, provided that n_i is a constant. Otherwise, _ALPHA_ is assigned the special missing value *V*.
3. If the limits are probability limits, the value of _SIGMAS_ is computed as $(\text{\_UCLU\_} - \text{\_U\_})/\sqrt{\text{\_U\_}/\text{\_LIMITN\_}}$, provided that n_i is a constant. Otherwise, _SIGMAS_ is assigned the special missing value *V*.
4. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the UCHART statement. For an example, see "Saving Control Limits" on page 1654.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables are saved:

- the *subgroup-variable*
- a subgroup number of nonconformities per unit variable named by *process* suffixed with *U*
- a subgroup sample size variable named by *process* suffixed with *N*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the UCHART statement. For example, consider the following statements:

```
proc shewhart data=fabric;
   uchart (flaws ndefects)*lot / outhistory=summary
                                 subgroupn = 10;
run;
```

The data set SUMMARY contains the variables LOT, FLAWSU, FLAWSN, NDEFCTSU, and NDEFCTSN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Nonconformities per Unit" on page 1660. Note that an OUTHISTORY= data set created with the UCHART statement can be used as a HISTORY= data set by either the CCHART statement or the UCHART statement.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables are saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on u chart
LCLU	lower control limit for number of nonconformities per unit
LIMITN	nominal sample size associated with the control limits
SIGMAS	multiple (k) of the standard error associated with the control limits
subgroup	values of the subgroup variable
SUBU	subgroup number of nonconformities per unit
SUBN	subgroup sample size
TESTS	tests for special causes signaled on u chart
UCLU	upper control limit for number of nonconformities per unit
VAR	*process* specified in the UCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved, depending on how the control limits are defined (with the ALPHA= or SIGMAS= option, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The k^{th} character of a value of _TESTS_ is k if Test k is positive at that subgroup. For example, if you request the first four tests (the ones appropriate for u charts)

and Tests 2 and 4 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, a 4 for the fourth character, and blanks for the other six characters.

3. The variables _EXLIM_ and _TESTS_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1654.

ODS Tables

The following table summarizes the ODS tables that you can request with the UCHART statement.

Table 48.23. ODS Tables Produced with the UCHART Statement

Table Name	Description	Options
UCHART	u chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

Input Data Sets

DATA= Data Set

You can read defect counts for subgroup samples from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the UCHART statement must be a SAS variable in the data set. This variable provides the defect count (number of nonconformities) for subgroup samples indexed by the *subgroup-variable*. The *subgroup-variable*, specified in the UCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. The data set should contain one observation per subgroup. When you use a DATA= data set with the UCHART statement, the SUBGROUPN= option (which specifies the number of inspection units per subgroup) is required. Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected

groups of observations (referred to as *phases*) with the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 1936).

For an example of a DATA= data set, see "Creating u Charts from Defect Count Data" on page 1652.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set CONLIMS:*

```
proc shewhart data=info limits=conlims;
   uchart defects*lot / subgroupn = 10;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLU_, _U_, and _UCLU_, which specify the control limits
- the variable _U_, which is used to calculate the control limits (see page 1674)

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are `ESTIMATE` and `STANDARD`.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1656.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This allows you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART procedure or to read output data sets created with SAS summarization procedures.

A HISTORY= data set used with the UCHART statement must contain the following variables:

- *subgroup-variable*

*In Release 6.09 and in earlier releases, it is necessary to specify the READLIMITS option.

- subgroup number of nonconformities per unit variable for each *process*
- subgroup sample size variable (number of units per subgroup) for each *process*

The names of the variables containing the number of nonconformities per unit and subgroup sample sizes must be the *process* name concatenated with the special suffix characters U and N, respectively. For example, consider the following statements:

```
proc shewhart history=summary;
   uchart (flaws ndefects)*lot;
run;
```

The data set SUMMARY must include the variables LOT, FLAWSU, FLAWSN, NDEFCTSU, and NDEFCTSN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (see "Displaying Stratification in Phases" on page 1936 for an example).

For an example of a HISTORY= data set, see "Creating u Charts from Nonconformities per Unit" on page 1657.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure or to create your own TABLE= data set. Because the SHEWHART procedure simply displays the information read from a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in Chapter 56, "Specialized Control Charts."

The following table lists the variables required in a TABLE= data set used with the UCHART statement:

Table 48.24. Variables Required in a TABLE= Data Set

Variable	Description
LCLU	lower control limit for nonconformities per unit
LIMITN	nominal sample size associated with the control limits
subgroup-variable	values of the *subgroup-variable*
SUBN	subgroup sample size
SUBU	subgroup number of nonconformities per unit
_U_	average number of nonconformities per unit
UCLU	upper control limit for nonconformities per unit

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1654.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical	DATA=	*process*
Vertical	HISTORY=	subgroup defects per unit variable
Vertical	TABLE=	_SUBU_

For an example, see "Labeling Axes" on page 1966.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples

This section provides advanced examples of the UCHART statement.

Example 48.1. Applying Tests for Special Causes

See SHWUEX1 in the SAS/QC Sample Library

This example illustrates how you can apply tests for special causes to make u charts more sensitive to special causes of variation.

A textile company inspects rolls of fabric for defects. The rolls are one meter wide and 30 meters long. The following statements create a SAS data set named FABRIC3, which contains the number of fabric defects for 20 rolls of fabric:

```
data fabric3;
   input roll defects @@;
   datalines;
 1   6    2   9    3  14    4  17
 5   3    6   8    7   9    8   2
 9  14   10   1   11   3   12   5
13   6   14   9   15  10   16  12
17  11   18   4   19   9   20   4
;
run;
```

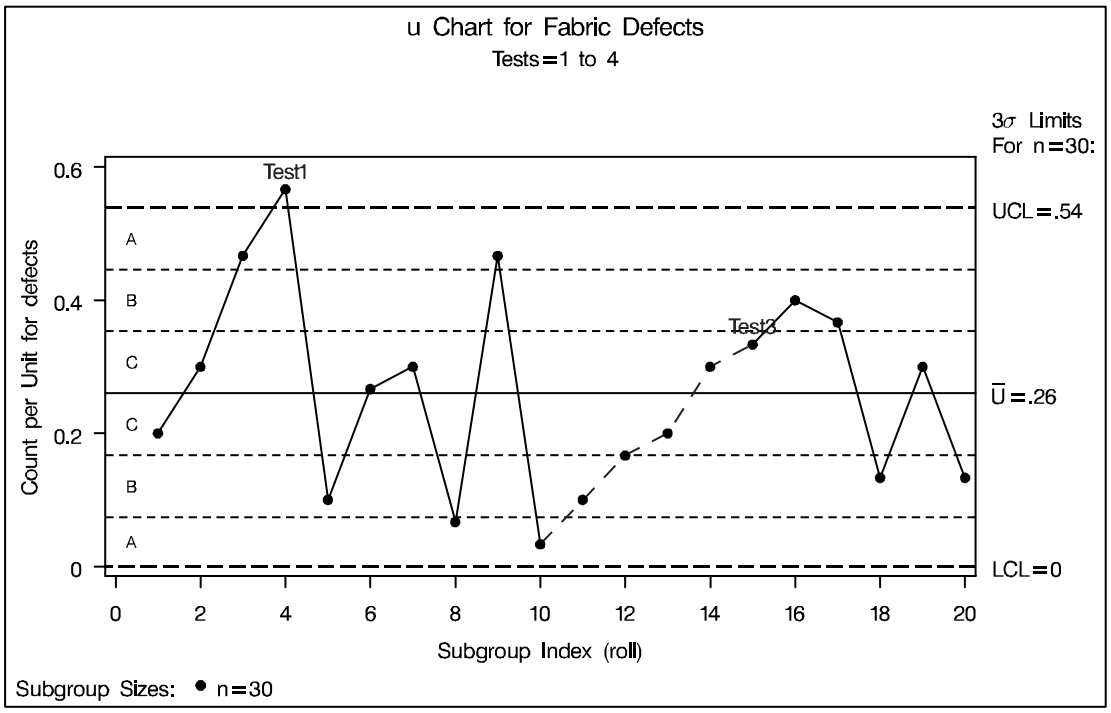
The following statements create a u chart and tabulate the information on the chart. The chart and tables are shown in Output 48.1.1 and Output 48.1.2.

```
title1 'u Chart for Fabric Defects';
title2 'Tests=1 to 4';
proc shewhart data=fabric3;
   uchart defects*roll / subgroupn = 30
                         tests     = 1 to 4
                         ltests    = 20
                         zonelabels
                         tabletests;
run;
```

The TESTS= option requests Tests 1, 2, 3, and 4, which are described in Chapter 55, "Tests for Special Causes." Only Tests 1, 2, 3, and 4 are recommended for u charts. The ZONELABELS option requests the zone lines, which are used to define the tests, and displays labels for the zones. The LTESTS= option specifies the line type used to connect the points in a pattern for a test that is signaled. The TABLETESTS option requests a table of the values of u_i and the control limits, together with a column indicating the subgroups at which the tests are positive.

Output 48.1.1 and Output 48.1.2 indicate that Test 1 is positive for Roll 4 and Test 3 is positive at Roll 15.

Output 48.1.1. Tests for Special Causes Displayed on u Chart



Output 48.1.2. Tabular Form of u Chart

u Chart for Fabric Defects
Tests=1 to 4

u Chart Summary for defects

roll	Subgroup Sample Size	-3 Sigma Limits with n=30 for Count per Unit- Lower Limit	Subgroup Count per Unit	Upper Limit	Special Tests Signaled
1	30.0000	0	0.20000000	0.53928480	
2	30.0000	0	0.30000000	0.53928480	
3	30.0000	0	0.46666667	0.53928480	
4	30.0000	0	0.56666667	0.53928480	1
5	30.0000	0	0.10000000	0.53928480	
6	30.0000	0	0.26666667	0.53928480	
7	30.0000	0	0.30000000	0.53928480	
8	30.0000	0	0.06666667	0.53928480	
9	30.0000	0	0.46666667	0.53928480	
10	30.0000	0	0.03333333	0.53928480	
11	30.0000	0	0.10000000	0.53928480	
12	30.0000	0	0.16666667	0.53928480	
13	30.0000	0	0.20000000	0.53928480	
14	30.0000	0	0.30000000	0.53928480	
15	30.0000	0	0.33333333	0.53928480	3
16	30.0000	0	0.40000000	0.53928480	
17	30.0000	0	0.36666667	0.53928480	
18	30.0000	0	0.13333333	0.53928480	
19	30.0000	0	0.30000000	0.53928480	
20	30.0000	0	0.13333333	0.53928480	

Example 48.2. Specifying a Known Expected Number of Nonconformities

See SHWUEX2 in the SAS/QC Sample Library

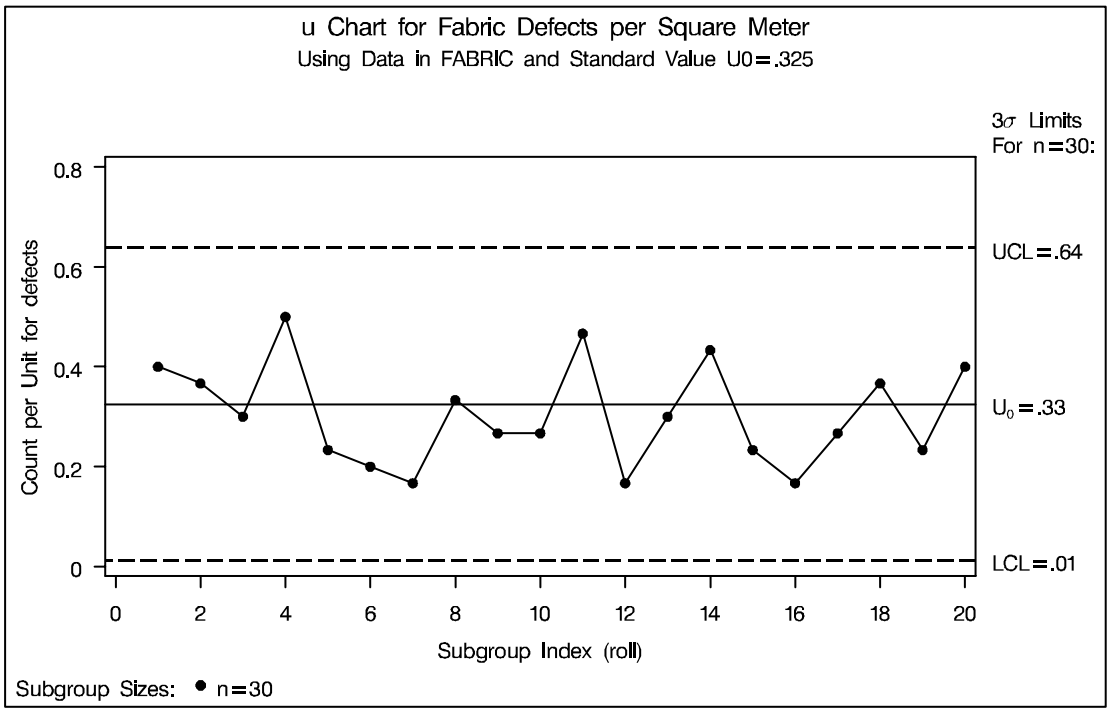
This example illustrates how you can create a u chart based on a known (standard) value u_0 for the expected number of nonconformities per unit.

A u chart is used to monitor the number of defects per square meter of fabric. The defect counts are provided as values of the variable DEFECTS in the data set FABRIC (see page 1652). Based on previous testing, it is known that $u_0 = 0.325$. The following statements create a u chart with control limits derived from this value:

```
title  'u Chart for Fabric Defects per Square Meter';
title2 'Using Data in FABRIC and Standard Value U0=.325';
proc shewhart data=fabric;
   uchart defects*roll / subgroupn = 30
                         u0        = 0.325
                         usymbol   = u0;
run;
```

The chart is shown in Output 48.2.1. The U0= option specifies u_0, and the USYMBOL= option requests a label for the central line indicating that the line represents a standard value.

Output 48.2.1. A u Chart with Standard Value u_0



Since all the points lie within the control limits, the process is in statistical control.

Alternatively, you can specify u_0 as the value of the variable _U_ in a LIMITS= data set, as follows:

```
data tlimits;
   length _subgrp_ _var_ _type_ $8;
   _u_      = .325;
   _limitn_ = 30;
   _type_   = 'STANDARD';
   _subgrp_ = 'roll';
   _var_    = 'defects';

proc shewhart data=fabric limits=tlimits;
   uchart defects*roll / subgroupn=30
                         usymbol  =u0;
run;
```

The chart produced by these statements is identical to the one in Output 48.2.1. For further details, see "LIMITS= Data Set" on page 1679.

Example 48.3. Creating u Charts for Varying Numbers of Units

See SHWUEX3 in the SAS/QC Sample Library

In the fabric manufacturing process described in "Creating u Charts from Defect Count Data" on page 1652, each roll of fabric is 30 meters long, and an inspection unit is defined as one square meter. Thus, there are 30 inspection units in each subgroup sample. Suppose now that the length of each piece of fabric varies. The following statements create a SAS data set (FABRICS2) that contains the number of fabric defects and size (in square meters) of 25 pieces of fabric:

```
data fabrics2;
   input roll defects sqmeters @@;
datalines;
 1  7 30.0   2 11 27.6   3 15 30.4   4  6 34.8   5 11 26.0
 6 15 28.6   7  5 28.0   8 10 30.2   9  8 28.2  10  3 31.4
11  3 30.3  12 14 27.8  13  3 27.0  14  9 30.0  15  7 32.1
16  6 34.8  17  7 26.5  18  5 30.0  19 14 31.3  20 13 31.6
21 11 29.4  22  6 28.6  23  6 27.5  24  9 32.6  25 11 31.7
;
run;
```

A partial listing of FABRICS2 is shown in Output 48.3.1.

Output 48.3.1. The Data Set FABRICS2

Number of Fabric Defects

roll	defects	sqmeters
1	7	30.0
2	11	27.6
3	15	30.4
.	.	.
.	.	.
.	.	.

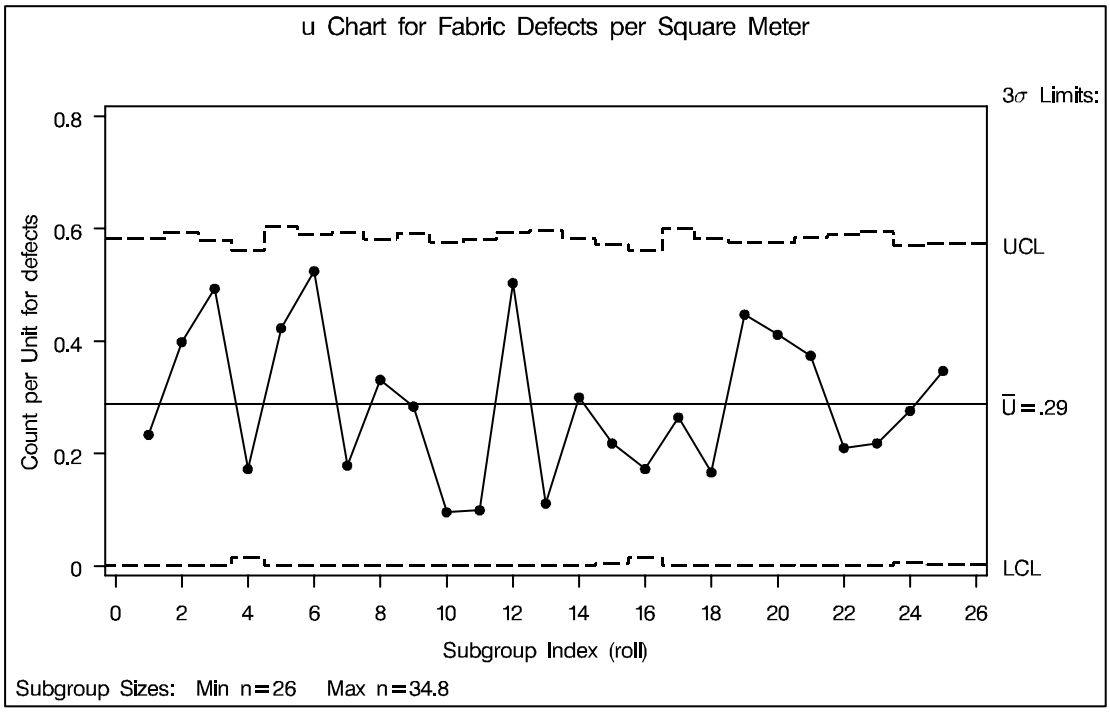
The variable ROLL contains the roll number, the variable DEFECTS contains the number of defects in each piece of fabric, and the variable SQMETERS contains the size of each piece.

The following statements request a u chart for the number of defects per square meter:

```
title 'u Chart for Fabric Defects per Square Meter';
proc shewhart data=fabrics2;
   uchart defects*roll / subgroupn = sqmeters
                         outlimits = flimits;
run;
```

The u chart is shown in Output 48.3.2, and the data set FLIMITS is listed in Output 48.3.3.

Output 48.3.2. A u Chart with Varying Number of Units per Subgroup



Note that the control limits vary with the number of units per subgroup (subgroup sample size). The legend in the lower left corner indicates the minimum and maximum subgroup sample sizes.

Output 48.3.3. The Control Limits Data Set FLIMITS

Control Limits for Fabric Defects

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLU_	_U_	_UCLU_
defects	roll	ESTIMATE	V	V	3	V	0.28805	V

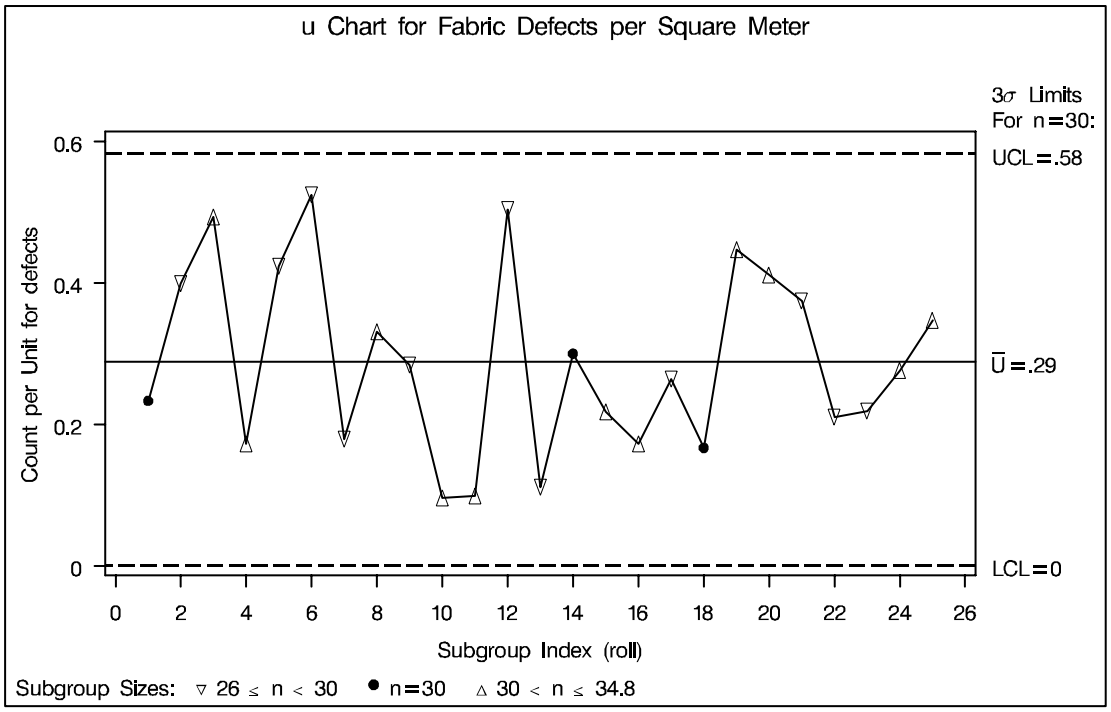
Output 48.3.3 shows that the variables _LIMITN_, _ALPHA_, _LCLU_, and _UCLU_ have the special missing value *V*, indicating that these variables vary with the sample size.

The following statements request a u chart with a fixed sample size of 30.0 for the control limits. In other words, the control limits are computed as if each piece of fabric were 30 meters long.

```
title 'u Chart for Fabric Defects per Square Meter';
proc shewhart data=fabrics2;
   uchart defects*roll / subgroupn = sqmeters
                         outlimits = flimits2
                         limitn    = 30
                         alln
                         nmarkers;
run;
```

The ALLN option specifies that points are to be displayed for all subgroups, regardless of their sample size. By default, when you specify the LIMITN= option, only points for subgroups whose sample size matches the LIMITN= value are displayed. The NMARKERS option requests special symbols that identify points for which the subgroup sample size differs from the nominal sample size of 30. The chart is shown in Output 48.3.4.

Output 48.3.4. Control Limits Based on Fixed Subgroup Sample Size



In Output 48.3.4, no points lie outside the control limits, indicating that the process is in control. However, you should be careful when interpreting charts that use a nominal sample size, since the fixed control limits based on this value are only an approximation. Output 48.3.5 lists the data set FLIMITS2, which contains the fixed control limits displayed in Output 48.3.4.

Output 48.3.5. The Fixed Control Limits Data Set FLIMITS2

Fixed Control Limits for Fabric Defects

VAR	_SUBGRP_	_TYPE_	_LIMIT_	_ALPHA_	_SIGMAS_	_LCL_	_U_	_UCL_
defects	roll	ESTIMATE	30	.002621618	3	0	0.28805	0.58201

Chapter 49
XCHART Statement

Chapter Contents

OVERVIEW 1691

GETTING STARTED 1692
- Creating Charts for Means from Raw Data 1692
- Creating Charts for Means from Subgroup Summary Data 1694
- Saving Summary Statistics 1697
- Saving Control Limits 1699
- Reading Preestablished Control Limits 1701

SYNTAX 1703
- Summary of Options 1704

DETAILS 1714
- Constructing Charts for Means 1714
- Output Data Sets 1716
- ODS Tables 1719
- Input Data Sets 1719
- Methods for Estimating the Standard Deviation 1723
- Axis Labels 1725
- Missing Values 1726

EXAMPLES 1726
- Example 49.1. Applying Tests for Special Causes 1726
- Example 49.2. Estimating the Process Standard Deviation 1728
- Example 49.3. Plotting OC Curves for Mean Charts 1731
- Example 49.4. Computing Process Capability Indices 1732

Chapter 49
XCHART Statement

Overview

The XCHART statement creates an $\bar{X}$ chart for subgroup means, which is used to analyze the central tendency of a process.

You can use options in the XCHART statement to

- compute control limits from the data based on a multiple of the standard error of the plotted means or as probability limits
- tabulate subgroup sample sizes, subgroup means, control limits, and other information
- save control limits in an output data set
- save subgroup sample sizes and subgroup means in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify one of several methods for estimating the process standard deviation
- specify whether subgroup standard deviations or subgroup ranges are used to estimate the process standard deviation
- specify a known (standard) process mean and standard deviation for computing control limits
- create a secondary chart that displays a time trend removed from the data (see "Displaying Trends in Process Data" on page 1957)
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

Note: When working with variables data, you should analyze the variability of the process as well as its central tendency. You can use the XRCHART statement or the XSCHART statement in the SHEWHART procedure for this purpose.

Getting Started

This section introduces the XCHART statement with simple examples that illustrate the most commonly used options. Complete syntax for the XCHART statement is presented in the "Syntax" section on page 1703, and advanced examples are given in the "Examples" section on page 1726.

Creating Charts for Means from Raw Data

See SHWXCHR in the SAS/QC Sample Library

Subgroup samples of five parts are taken from the manufacturing process at regular intervals, and the width of a critical gap in each part is measured in millimeters. The following statements create a SAS data set named PARTGAPS, which contains the gap width measurements for 21 samples:

```
data partgaps;
   input sample @;
   do i=1 to 5;
      input partgap @;
      output;
      end;
   drop i;
   label partgap='Gap Width'
         sample ='Sample Index';
   datalines;
 1 255 270 268 290 267
 2 260 240 265 262 263
 3 238 236 260 250 256
 4 260 242 281 254 263
 5 268 260 279 289 269
 6 270 249 265 253 263
 7 280 260 256 256 243
 8 229 266 250 243 252
 9 250 270 245 273 262
10 248 258 247 266 256
11 280 251 252 270 287
12 245 253 243 279 245
13 268 260 289 275 273
14 264 286 275 271 279
15 271 257 263 247 247
16 291 250 273 265 266
17 228 253 240 260 264
18 270 260 269 245 276
19 259 257 246 271 257
20 252 244 230 266 248
21 254 251 239 233 263
;
run;
```

A partial listing of PARTGAPS is shown in Figure 49.1.

The Data Set PARTGAPS

sample	partgap
1	255
1	270
1	268
1	290
1	267
2	260
2	240
2	265
2	262
2	263
.	.
.	.
.	.

Figure 49.1. Partial Listing of the Data Set PARTGAPS

The data set PARTGAPS is said to be in "strung-out" form, since each observation contains the sample number and gap width measurement for a single part. The first five observations contain the gap widths for the first sample, the second five observations contain the gap widths for the second sample, and so on. Because the variable SAMPLE classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable PARTGAP contains the gap width measurements and is referred to as the *process variable* (or *process* for short).

The within-subgroup variability of the gap widths is known to be stable. You can use an $\bar{X}$ chart to determine whether their mean level is in control. The following statements create the $\bar{X}$ chart shown in Figure 49.2:

```
title 'Mean Chart for Gap Widths';
proc shewhart data=partgaps;
   xchart partgap*sample;
run;
```

This example illustrates the basic form of the XCHART statement. After the keyword XCHART, you specify the *process* to analyze (in this case, PARTGAP) followed by an asterisk and the *subgroup-variable* (SAMPLE).

The input data set is specified with the DATA= option in the PROC SHEWHART statement.

For more information on the SYMBOL statement, refer to *SAS/GRAPH Software: Reference.*

Each point on the $\bar{X}$ chart represents the average (mean) of the measurements for a particular sample. For instance, the mean plotted for the first sample is

$$\frac{255 + 270 + 268 + 290 + 267}{5} = 270$$

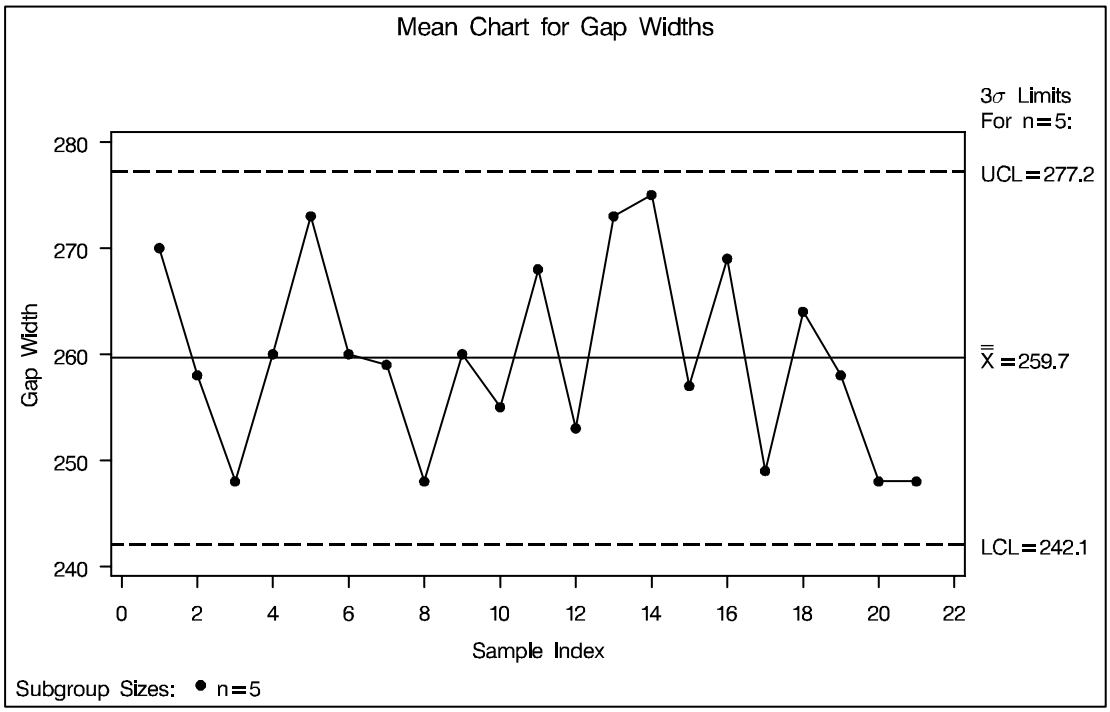


Figure 49.2. $\bar{X}$ Chart for Gap Width Data

Since all of the subgroup means lie within the control limits, it can be concluded that the mean level of the process is in statistical control.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 49.23 on page 1715. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1701.

For computational details, see "Constructing Charts for Means" on page 1714. For details on reading raw measurements, see "DATA= Data Set" on page 1719.

Creating Charts for Means from Subgroup Summary Data

See SHWXCHR in the SAS/QC Sample Library

The previous example illustrates how you can create $\bar{X}$ charts using raw data (process measurements). However, in many applications, the data are provided as subgroup summary statistics. This example illustrates how you can use the XCHART statement with data of this type.

The following data set (PARTS) provides the data from the preceding example in summarized form:

```
data parts;
   input sample partgapx partgapr;
   partgapn=5;
   label partgapx='Mean of Gap Width'
         sample  ='Sample Index';
   datalines;
 1  270  35
 2  258  25
 3  248  24
 4  260  39
 5  273  29
 6  260  21
 7  259  37
 8  248  37
 9  260  28
10  255  19
11  268  36
12  253  36
13  273  29
14  275  22
15  257  24
16  269  41
17  249  36
18  264  31
19  258  25
20  248  36
21  248  30
;
run;
```

A partial listing of PARTS is shown in Figure 49.3. There is exactly one observation for each subgroup (note that the subgroups are still indexed by SAMPLE). The variable PARTGAPX contains the subgroup means, the variable PARTGAPR contains the subgroup ranges, and the variable PARTGAPN contains the subgroup sample sizes (these are all five).

The Data Set PARTS

sample	partgapx	partgapr	partgapn
1	270	35	5
2	258	25	5
3	248	24	5
4	260	39	5
5	273	29	5
.	.	.	.
.	.	.	.
.	.	.	.

Figure 49.3. The Summary Data Set PARTS

You can read this data set by specifying it as a HISTORY= data set in the PROC SHEWHART statement, as follows:

```
title 'Mean Chart for Gap Width';
proc shewhart history=parts lineprinter;
   xchart partgap*sample='*';
run;
```

The resulting $\bar{X}$ chart is shown in Figure 49.4. Since the LINEPRINTER option is specified in the PROC SHEWHART statement, line printer output is produced. The asterisk (*) specified in single quotes after the *subgroup-variable* indicates the character used to plot points. This character must follow an equal sign.

Note that PARTGAP is *not* the name of a SAS variable in the data set but is, instead, the common prefix for the names of the three SAS variables PARTGAPX, PARTGAPR, and PARTGAPN. The suffix characters *X*, *R*, and *N* indicate *mean*, *range*, and *sample size*, respectively. Thus, you can specify three subgroup summary variables in a HISTORY= data set with a single name (PARTGAP), which is referred to as the *process*. The name SAMPLE specified after the asterisk is the name of the *subgroup-variable*.

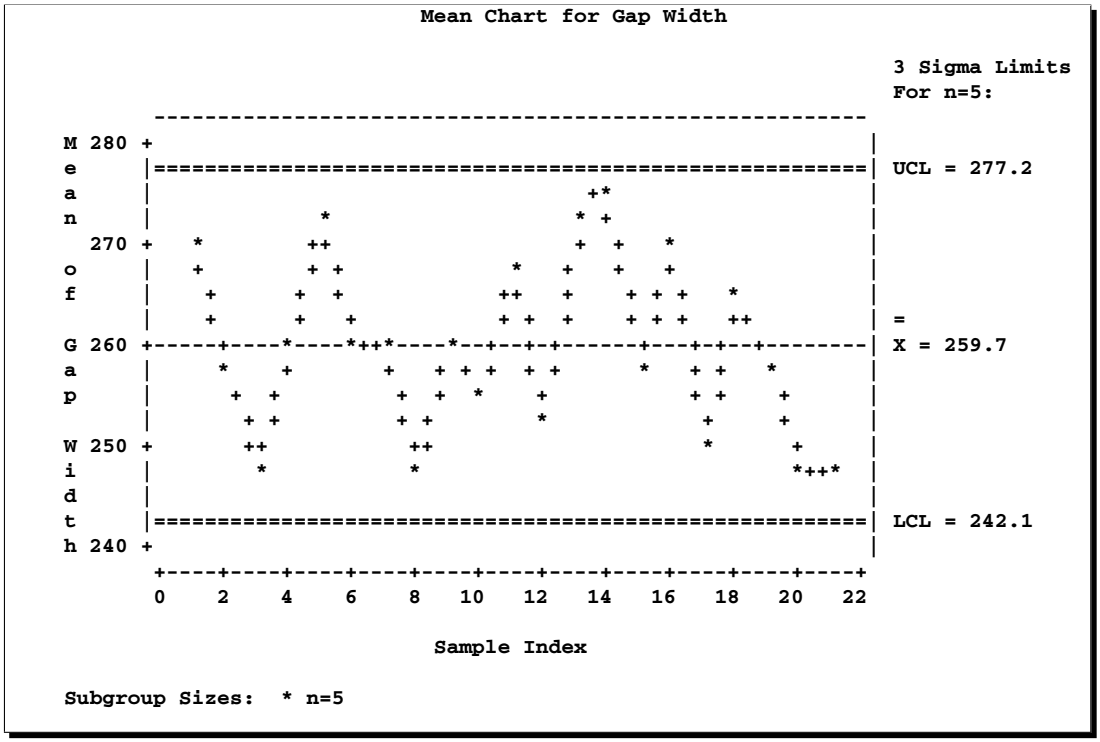


Figure 49.4. $\bar{X}$ Chart from the Summary Data Set PARTS

In general, a HISTORY= input data set used with the XCHART statement must contain the following variables:

- subgroup variable
- subgroup mean variable
- either a subgroup range variable or a subgroup standard deviation variable
- subgroup sample size variable

Furthermore, the names of the subgroup mean, range (or standard deviation), and sample size variables must begin with the *process* name specified in the XCHART statement and end with the special suffix characters *X*, *R* (or *S*), and *N*, respectively. If the names do not follow this convention, you can use the RENAME option in the PROC SHEWHART statement to rename the variables for the duration of the SHEWHART procedure step (see page 1743).

If you specify the STDDEVIATIONS option in the XCHART statement, the HISTORY= data set must contain a subgroup standard deviation variable; otherwise, the HISTORY= data set must contain a subgroup range variable. The STDDEVIATIONS option specifies that the estimate of the process standard deviation σ is to be calculated from subgroup standard deviations rather than subgroup ranges. For example, in the following statements, the data set PARTS2 must contain a subgroup standard deviation variable named PARTGAPS:

```
title 'Mean Chart for Gap Width';
proc shewhart history=parts2;
   xchart partgap*sample='*' / stddeviations;
run;
```

Options such as STDDEVIATIONS are specified after the slash (/) in the XCHART statement. A complete list of options is presented in the "Syntax" section on page 1703.

In summary, the interpretation of *process* depends on the input data set.

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names of the variables containing the summary statistics.

For more information, see "HISTORY= Data Set" on page 1721.

Saving Summary Statistics

See SHWXCHR in the SAS/QC Sample Library

In this example, the XCHART statement is used to create a summary data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read measurements from the data set PARTGAPS and create a summary data set named GAPHIST:

```
proc shewhart data=partgaps;
   xchart partgap*sample / outhistory = gaphist
                           nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 49.2.

Figure 49.5 contains a partial listing of GAPHIST.

Summary Data Set for Gap Widths

sample	partgap X	partgap R	partgap N
1	270	35	5
2	258	25	5
3	248	24	5
4	260	39	5
5	273	29	5
.	.	.	.
.	.	.	.
.	.	.	.

Figure 49.5. The Summary Data Set GAPHIST

There are four variables in the data set GAPHIST.

- SAMPLE contains the subgroup index.
- PARTGAPX contains the subgroup means.
- PARTGAPR contains the subgroup ranges.
- PARTGAPN contains the subgroup sample sizes.

Note that the summary statistic variables are named by adding the suffix characters *X*, *R*, and *N* to the *process* PARTGAP specified in the XCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

If you specify the STDDEVIATIONS option, the OUTHISTORY= data set includes a subgroup standard deviation variable rather than a subgroup range variable, as demonstrated by the following statements:

```
proc shewhart data=partgaps;
   xchart partgap*sample / outhistory = gaphist2
                           stddeviations
                           nochart;
run;
```

Figure 49.6 contains a partial listing of GAPHIST2.

Summary Data Set with Subgroup Standard Deviations

sample	partgap X	partgap S	partgap N
1	270	12.6293	5
2	258	10.2225	5
3	248	10.6771	5
4	260	14.2302	5
5	273	11.2027	5
.	.	.	.
.	.	.	.
.	.	.	.

Figure 49.6. The Summary Data Set GAPHIST2

The variable PARTGAPS, which contains the subgroup standard deviations, is named by adding the suffix character *S* to the *process* PARTGAP.

For more information, see "OUTHISTORY= Data Set" on page 1717.

Saving Control Limits

See SHWXCHR in the SAS/QC Sample Library

You can save the control limits for an $\bar{X}$ chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1701) or modify the limits with a DATA step program.

The following statements read measurements from the data set PARTGAPS (see page 1692) and save the control limits displayed in Figure 49.2 in a data set named GAPLIM:

```
proc shewhart data=partgaps;
   xchart partgap*sample / outlimits = gaplim
                           nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. The data set GAPLIM is listed in Figure 49.7.

Control Limits for Gap Width Measurements

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_
partgap	sample	ESTIMATE	5	.002699796	3

LCLX	_MEAN_	_UCLX_	_LCLR_	_R_	_UCLR_	_STDDEV_
242.087	259.667	277.246	0	30.4762	64.4419	13.1028

Figure 49.7. The Data Set GAPLIM Containing Control Limit Information

The data set GAPLIM contains one observation with the limits for *process* PARTGAP. The variables _LCLX_ and _UCLX_ contain the lower and upper control limits for the means, and the variable _MEAN_ contains the central line. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values.

The variables _LCLR_, _R_, and _UCLR_ are not used to create $\bar{X}$ charts, but they are included so the data set GAPLIM can be used to create an R chart; see Chapter 50, "XRCHART Statement." If you specify the STDDEVIATIONS option in the XCHART statement, the variables _LCLS_, _S_, and _UCLS_ are included in the OUTLIMITS= data set. These variables can be used to create an s chart; see Chapter 51, "XSCHART Statement." For more information, see "OUTLIMITS= Data Set" on page 1716.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=partgaps;
   xchart partgap*sample / outtable=gtable
                           nochart;
run;
```

The data set GTABLE is listed in Figure 49.8.

Summary Statistics and Control Limit Information

VAR	sample	_SIGMAS_	_LIMITN_	_SUBN_	_LCLX_	_SUBX_	_MEAN_	_UCLX_	_EXLIM_
partgap	1	3	5	5	242.087	270	259.667	277.246	
partgap	2	3	5	5	242.087	258	259.667	277.246	
partgap	3	3	5	5	242.087	248	259.667	277.246	
partgap	4	3	5	5	242.087	260	259.667	277.246	
partgap	5	3	5	5	242.087	273	259.667	277.246	
partgap	6	3	5	5	242.087	260	259.667	277.246	
partgap	7	3	5	5	242.087	259	259.667	277.246	
partgap	8	3	5	5	242.087	248	259.667	277.246	
partgap	9	3	5	5	242.087	260	259.667	277.246	
partgap	10	3	5	5	242.087	255	259.667	277.246	
partgap	11	3	5	5	242.087	268	259.667	277.246	
partgap	12	3	5	5	242.087	253	259.667	277.246	
partgap	13	3	5	5	242.087	273	259.667	277.246	
partgap	14	3	5	5	242.087	275	259.667	277.246	
partgap	15	3	5	5	242.087	257	259.667	277.246	
partgap	16	3	5	5	242.087	269	259.667	277.246	
partgap	17	3	5	5	242.087	249	259.667	277.246	
partgap	18	3	5	5	242.087	264	259.667	277.246	
partgap	19	3	5	5	242.087	258	259.667	277.246	
partgap	20	3	5	5	242.087	248	259.667	277.246	
partgap	21	3	5	5	242.087	248	259.667	277.246	

Figure 49.8. The Data Set GTABLE

This data set contains one observation for each subgroup sample. The variables _SUBX_ and _SUBN_ contain the subgroup means and sample sizes. The variables _LCLX_ and _UCLX_ contain the lower and upper control limits, and the variable _MEAN_ contains the central line. The variables _VAR_ and SAMPLE contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1718.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read GTABLE and display an $\bar{X}$ chart (not shown here) identical to the chart in Figure 49.2:

```
title 'Mean Chart for Gap Widths';
proc shewhart table=gtable;
   xchart partgap*sample;
label _SUBX_ = 'Gap Width';
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see Chapter 56, "Specialized Control Charts,").

For more information, see "TABLE= Data Set" on page 1722.

Reading Preestablished Control Limits

In the previous example, the OUTLIMITS= data set GAPLIM saved control limits computed from the measurements in PARTGAPS. This example shows how these limits can be applied to new data provided in the following data set:

See SHWXCHR in the SAS/QC Sample Library

```
data gaps2;
   input sample @;
   do i=1 to 5;
      input partgap @;
      output;
      end;
   drop i;
   datalines;
22 287 265 248 263 271
23 267 253 285 251 271
24 249 252 277 269 241
25 243 248 263 282 261
26 287 266 256 278 242
27 251 262 243 274 245
28 256 245 244 243 272
29 262 247 252 277 266
30 244 269 263 278 261
31 245 264 246 242 273
32 272 257 277 265 241
33 251 249 240 260 261
34 289 277 275 273 261
35 267 286 275 261 272
36 266 256 247 255 241
37 291 267 267 252 262
```

```
38 258 245 264 245 281
39 277 267 241 272 244
40 252 267 272 245 252
41 243 241 245 263 248
;
run;
```

The following statements create an $\bar{X}$ chart for the data in GAPS2 using the control limits in GAPLIM:

```
title 'Mean Chart for Gap Widths';
proc shewhart data=gaps2 limits=gaplim;
   xchart partgap*sample;
run;
```

The chart is shown in Figure 49.9.

The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default,* this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name PARTGAP
- the value of _SUBGRP_ matches the *subgroup-variable* name SAMPLE

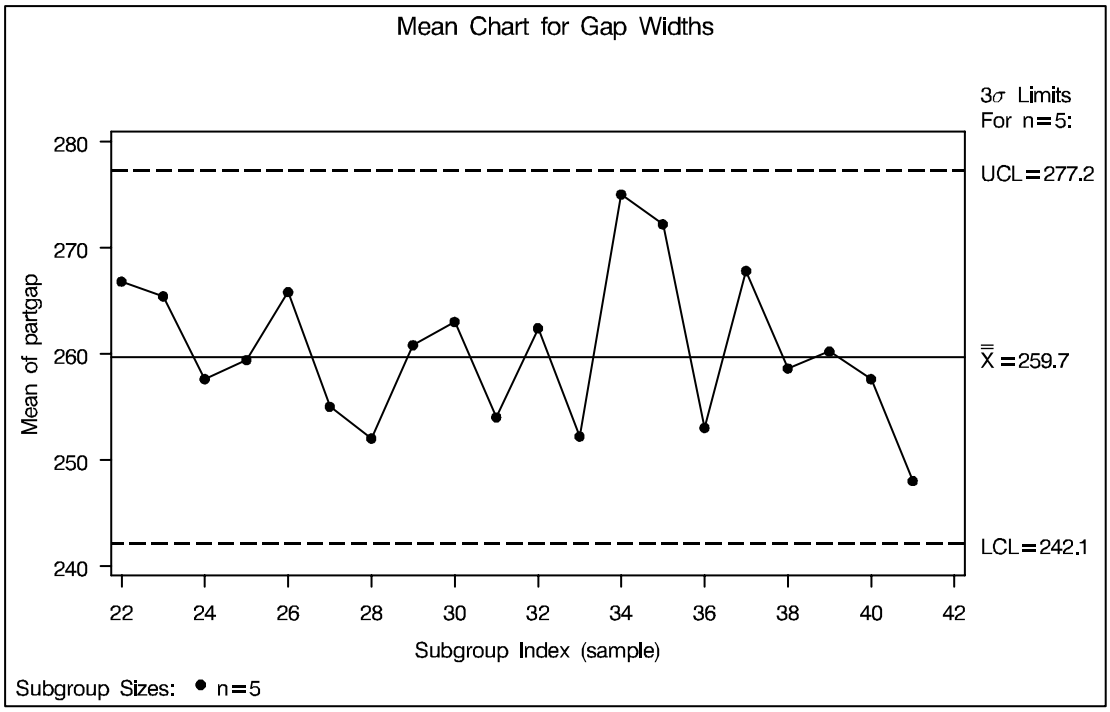


Figure 49.9. $\bar{X}$ Chart for Second Set of Gap Width Data

*In Release 6.09 and in earlier releases, it is also necessary to specify the READLIMITS option to read control limits from a LIMITS= data set.

The chart indicates that the process is in control, since all the means lie within the control limits.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1720 for details concerning the variables that you must provide.

Syntax

The basic syntax for the XCHART statement is as follows:

XCHART *process***subgroup-variable* **;**

The general form of this syntax is as follows:

XCHART (*processes***)****subgroup-variable* <**(***block-variables* **)** >
< =*symbol-variable* | =*'character'* > < **/** *options* >**;**

You can use any number of XCHART statements in the SHEWHART procedure. The components of the XCHART statement are described as follows.

process
processes
identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating Charts for Means from Raw Data" on page 1692.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating Charts for Means from Subgroup Summary Data" on page 1694.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1699.

A *process* is required. If you specify more than one process, enclose the list in parentheses. For example, the following statements request distinct $\bar{X}$ charts for WEIGHT, LENGTH, and WIDTH:

```
proc shewhart data=measures;
   xchart (weight length width)*day;
run;
```

subgroup-variable
is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding XCHART statement, DAY is the subgroup variable. For details, see "Subgroup Variables" on page 1771.

block-variables
are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 1932 for an example.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the means.

- If you produce a chart on a line printer, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce a chart on a graphics device, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOLn statements. See "Displaying Stratification in Levels of a Classification Variable" on page 1931 for an example.

character
specifies a plotting character for charts produced on line printers. For example, the following statements create an $\bar{X}$ chart using an asterisk (*) to plot the points:

```
proc shewhart data=values;
   xchart weight*day='*';
run;
```

options
enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The "Summary of Options" section, which follows, lists all options by function. Chapter 53, "Dictionary of Options," describes each option in detail.

Summary of Options

The following tables list the XCHART statement options by function. For complete descriptions, see Chapter 53, "Dictionary of Options."

Table 49.1. Tabulation Options

TABLE	creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with a column indicating which tests for special causes are positive

Note that specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.

Table 49.2. Options for Specifying Tests for Special Causes

NO3SIGMACHECK	enables tests to be applied with control limits other than 3σ limits
TESTS=*value-list*| *customized-pattern-list*	specifies tests for special causes
TEST2RUN=*n*	specifies length of pattern for Test 2
TEST3RUN=*n*	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL='*label*'| (*variable*)|*keyword*	provides labels for points where test is positive
TESTLABEL*n*='*label*'	specifies label for n^{th} test for special causes
TESTNMETHOD= STANDARDIZE	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
TESTRESET=*variable*	enables tests for special causes to be reset
ZONELABELS	adds labels A, B, and C to zone lines
ZONES	adds lines delineating zones A, B, and C
ZONEVALPOS=*n*	specifies position of ZONEVALUES labels
ZONEVALUES	labels zone lines with their values

Table 49.3. Graphical Options for Displaying Tests for Special Causes

CTESTLABBOX=*color*	specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=*color*|*test-color-list*	specifies color for labels indicating points where test is positive
CTESTSYMBOL=*color*	specifies color for symbol used to plot points where test is positive
CZONES=*color*	specifies color for lines and labels delineating zones A, B, and C
LTESTS=*linetype*	specifies type of line connecting points where test is positive
LZONES=*linetype*	specifies line type for lines delineating zones A, B, and C
TESTFONT=*font*	specifies software font for labels at points where test is positive
TESTHEIGHT=*value*	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=*symbol*	specifies plot symbol for points where test is positive
TESTSYMBOLHT=*value*	specifies symbol height for points where test is positive
WTESTS=*n*	specifies width of line connecting points where test is positive

Table 49.4. Line Printer Options for Displaying Tests for Special Causes

TESTCHAR='*character*'	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
ZONECHAR='*character*'	specifies character for lines that delineate zones for tests for special causes

Table 49.5. Reference Line Options

CHREF=*color*	specifies color for lines requested by HREF= and HREF2= options
CVREF=*color*	specifies color for lines requested by VREF= and VREF2= options
HREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on $\bar{X}$ chart
HREF2=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFCHAR='*character*'	specifies line character for HREF= and HREF2= lines
HREFDATA= *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on $\bar{X}$ chart
HREF2DATA= *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFLABELS= *'label1'...'labeln'*	specifies labels for HREF= lines
HREF2LABELS= *'label1'...'labeln'*	specifies labels for HREF2= lines
HREFLABPOS=*n*	specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=*linetype*	specifies line type for HREF= and HREF2= lines
LVREF=*linetype*	specifies line type for VREF= and VREF2= lines
NOBYREF	specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to vertical axis on $\bar{X}$ chart
VREF2=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to vertical axis on trend chart
VREFCHAR='*character*'	specifies line character for VREF= and VREF2= lines
VREFLABELS= *'label1'...'labeln'*	specifies labels for VREF= lines
VREF2LABELS= *'label1'...'labeln'*	specifies labels for VREF2= lines
VREFLABPOS=*n*	position of VREFLABELS= and VREF2LABELS= labels

Table 49.6. Block Variable Legend Options

BLOCKLABELPOS=*keyword*	specifies position of label for *block-variable* legend
BLOCKLABTYPE=*value*|*keyword*	specifies text size of *block-variable* legend
BLOCKPOS=*n*	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=*color-list*	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=*variable*|(*variables*)	specifies one or more variables whose values are colors for filling background of *block-variable* legend

Table 49.7. Axis and Axis Label Options

CAXIS=*color*	specifies color for axis lines and tick marks
CFRAME=*color*|(*color-list*)	specifies fill colors for frame for plot area
CTEXT=*color*	specifies color for tick mark values and axis labels
HAXIS=*values*|AXIS*n*	specifies major tick mark values for horizontal axis
HEIGHT=*value*	specifies height of axis label and axis legend text
HMINOR=*n*	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=*value*	specifies length of offset at both ends of horizontal axis
INTSTART=*value*	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=*n*	specifies thinning factor for tick mark labels on horizontal axis
SPLIT='*character*'	specifies splitting character for axis labels
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=*values*|AXIS*n*	specifies major tick mark values for vertical axis of $\bar{X}$ chart
VAXIS2=*values*|AXIS*n*	specifies major tick mark values for vertical axis of trend chart
VFORMAT=*format*	specifies format for primary vertical axis tick mark labels
VFORMAT2=*format*	specifies format for secondary vertical axis tick mark labels
VMINOR=*n*	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=*value*	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis for primary chart
VZERO2	forces origin to be included in vertical axis for secondary chart
WAXIS=*n*	specifies width of axis lines

Table 49.8. Options for Specifying Control Limits

ALPHA=*value*	requests probability limits for chart
LIMITN=*n*|VARYING	specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	computes control limits for each *process* from the data rather than a LIMITS= data set (Release 6.10 and later releases)
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEXES=ALL| '*label1*'...'*labeln*'	reads multiple sets of control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (Release 6.09 and earlier releases)
SIGMAS=*k*	specifies width of control limits in terms of multiple *k* of standard error of plotted means

Table 49.9. Options for Displaying Control Limits

CINFILL=*color*	specifies color for area inside control limits
CLIMITS=*color*	specifies color of control limits, central line, and related labels
LCLLABEL='*label*'	specifies label for lower control limit
LIMLABSUBCHAR= '*character*'	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=*linetype*	specifies line type for control limits
NDECIMAL=*n*	specifies number of digits to right of decimal place in default labels for control limits and central line
NOCTL	suppresses display of central line
NOLCL	suppresses display of lower control limit
NOLIMITLABEL	suppresses labels for control limits and central line
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOUCL	suppresses display of upper control limit
UCLLABEL='*string*'	specifies label for upper control limit
WLIMITS=*n*	specifies width for control limits and central line
XSYMBOL='*string*'| *keyword*	specifies label for central line

Table 49.10. Specification Limit Options

CIINDICES (ALPHA=*value* TYPE=*keyword*)	specifies α value and type for computing capability index confidence limits
LSL=*value-list*	specifies list of lower specification limits
TARGET=*value-list*	specifies list of target values
USL=*value-list*	specifies list of upper specification limits

Table 49.11. Options for Plotting and Labeling Points

ALLLABEL=VALUE| (*variable*)	labels every point on $\bar{X}$ chart
ALLLABEL2=VALUE| (*variable*)	labels every point on trend chart
CLABEL=*color*	specifies color for labels
CCONNECT=*color*	specifies color for line segments that connect points on chart
CFRAMELAB=*color*	specifies fill color for frame around labeled points
CNEEDLES=*color*	specifies color for needles that connect points to central line
CONNECTCHAR= '*character*'	specifies character used to form line segments that connect points on chart
COUT=*color*	specifies color for portions of line segments that connect points outside control limits
COUTFILL=*color*	specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=*angle*	specifies angle at which labels are drawn
LABELFONT=*font*	specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=*value*	specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on chart
NOTRENDCONNECT	suppresses line segments that connect points on trend chart
OUTLABEL=VALUE| (*variable*)	labels points outside control limits
SYMBOLCHARS= '*characters*'	specifies characters indicating *symbol-variable*
SYMBOLLEGEND= NONE|*name*	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER= *keyword*	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=*n*	specifies width of needles

Table 49.12. Clipping Options

CCLIP=*color*	specifies color for plot symbol for clipped points
CLIPCHAR='*character*'	specifies plot character for clipped points
CLIPFACTOR=*value*	determines extent to which extreme points are clipped
CLIPLEGEND='*string*'	specifies text for clipping legend
CLIPLEGPOS=*keyword*	specifies position of clipping legend
CLIPSUBCHAR='*character*'	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=*symbol*	specifies plot symbol for clipped points
CLIPSYMBOLHT=*value*	specifies symbol marker height for clipped points

Table 49.13. Phase Options

CPHASELEG=*color*	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE='*string*'	specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=*value*|*keyword*	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES= ALL|'*label1*'...'*labeln*'	specifies *phases* to be read from an input data set

Table 49.14. Process Mean and Standard Deviation Options

MU0=*value*	specifies known value of μ_0 for process mean μ
SIGMA0=*value*	specifies known value σ_0 for process standard deviation σ
SMETHOD=*keyword*	specifies method for estimating process standard deviation σ
STDDEVIATIONS	specifies that estimate of process standard deviation σ is to be calculated from subgroup standard deviations
TYPE=*keyword*	identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set

Table 49.15. Options for Interactive Control Charts

HTML=(*variable*)	specifies a variable whose values are URLs to be associated with subgroups
HTML2=(*variable*)	specifies variable whose values are URLs to be associated with subgroups on secondary chart
HTML_LEGEND=(*variable*)	specifies a variable whose values are URLs to be associated with symbols in the symbol legend
WEBOUT=*SAS-data-set*	creates an OUTTABLE= data set with additional graphics coordinate data

Table 49.16. Input Data Set Options

MISSBREAK	specifies that observations with missing values are not to be processed

Table 49.17. Output Data Set Options

OUTHISTORY=*SAS-data-set*	creates output data set containing subgroup summary statistics
OUTINDEX='*string*'	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=*SAS-data-set*	creates output data set containing control limits
OUTTABLE=*SAS-data-set*	creates output data set containing subgroup summary statistics and control limits

Table 49.18. Grid Options

CGRID=*color*	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=*linetype*	specifies line type for grid requested with the ENDGRID option
LGRID=*linetype*	specifies line type for grid requested with the GRID option
WGRID=*n*	specifies width of grid lines

Table 49.19. Plot Layout Options

ALLN	plots means for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts for a process only when exceptions occur
INTERVAL=*keyword*	natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=*n*	maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=*n*	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=*n*	specifies number of pages or screens to be used to display chart
TRENDVAR=*variable*|(*variable-list*)	specifies list of trend variables
YPCT1=*value*	specifies length of vertical axis on $\bar{X}$ chart as a percentage of sum of lengths of vertical axes for $\bar{X}$ and trend charts
ZEROSTD	displays $\bar{X}$ chart regardless of whether $\hat{\sigma} = 0$

Table 49.20. Graphical Enhancement Options

ANNOTATE=*SAS-data-set*	specifies annotate data set that adds features to $\bar{X}$ chart
ANNOTATE2=*SAS-data-set*	specifies annotate data set that adds features to trend chart
DESCRIPTION='*string*'	specifies string that appears in the description field of the PROC GREPLAY master menu for $\bar{X}$ chart
FONT=*font*	specifies software font for labels and legends on charts
NAME='*string*'	specifies name that appears in the name field of the PROC GREPLAY master menu for $\bar{X}$ chart
PAGENUM='*string*'	specifies the form of the label used in pagination
PAGENUMPOS=*keyword*	specifies the position of the page number requested with the PAGENUM= option
WTREND=*n*	specifies width of line segments connecting points on trend chart

Table 49.21. Star Options

CSTARCIRCLES=*color*	specifies color for STARCIRCLES= circles
CSTARFILL=*color*|*(variable)*	specifies color for filling stars
CSTAROUT=*color*	specifies outline color for stars exceeding inner or outer circles
CSTARS=*color*| *(variable)*	specifies color for outlines of stars
LSTARCIRCLES= *linetypes*	specifies line types for STARCIRCLES= circles
LSTARS=*linetype*|*(variable)*	specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=*value*	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=*value-list*	specifies reference circles for stars
STARINRADIUS=*value*	specifies inner radius of stars
STARLABEL=*keyword*	specifies vertices to be labeled
STARLEGEND=*keyword*	specifies style of legend for star vertices
STARLEGENDLAB='*label*'	specifies label for STARLEGEND= legend
STAROUTRADIUS=*value*	specifies outer radius of stars
STARSPEC=*value*|*SAS-data-set*	specifies method used to standardize vertex variables
STARSTART=*value*	specifies angle for first vertex
STARTYPE=*keyword*	specifies graphical style of star
STARVERTICES=*variable*|*(variables)*	superimposes star at each point on $\bar{X}$ chart
WSTARCIRCLES=*n*	specifies width of STARCIRCLES= circles
WSTARS=*n*	specifies width of STARVERTICES= stars

Table 49.22. Overlay Options

CCOVERLAY=*color-list*	specifies colors for primary chart overlay line segments
CCOVERLAY2=*color-list*	specifies colors for secondary chart overlay line segments
COVERLAY=*color-list*	specifies colors for primary chart overlay plots
COVERLAY2=*color-list*	specifies colors for secondary chart overlay plots
COVERLAYCLIP=*color*	specifies color for clipped points on overlays
LOVERLAY=*linetypes*	specifies line types for primary chart overlay line segments
LOVERLAY2=*linetypes*	specifies line types for secondary chart overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=*variable-list*	specifies variables to overlay on primary chart
OVERLAY2=*variable-list*	specifies variables to overlay on secondary chart
OVERLAY2HTML=*variable-list*	specifies URLS to associate with secondary chart overlay points
OVERLAY2ID=*variable-list*	specifies labels for secondary chart overlay points
OVERLAY2SYM=*symbol-list*	specifies symbols for secondary chart overlays
OVERLAY2SYMHT=*value-list*	specifies symbol heights for secondary chart overlays
OVERLAYCLIPSYM=*symbol*	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=*value*	specifies symbol height for clipped points on overlays
OVERLAYHTML=*variable-list*	specifies URLS to associate with primary chart overlay points
OVERLAYID=*variable-list*	specifies labels for primary chart overlay points
OVERLAYLEGLAB='*label*'	specifies label for overlay legend
OVERLAYSYM=*symbol-list*	specifies symbols for primary chart overlays
OVERLAYSYMHT=*value-list*	specifies symbol heights for primary chart overlays
WOVERLAY=*value-list*	specifies widths of primary chart overlay line segments
WOVERLAY2=*value-list*	specifies widths of secondary chart overlay line segments

Details

Constructing Charts for Means

The following notation is used in this section:

μ	process mean (expected value of the population of measurements)
σ	process standard deviation (standard deviation of the population of measurements)
$\bar{X}_i$	mean of measurements in i^{th} subgroup
R_i	range of measurements in i^{th} subgroup
n_i	sample size of i^{th} subgroup
N	number of subgroups
$\overline{\overline{X}}$	weighted average of subgroup means
z_p	$100p^{\text{th}}$ percentile of the standard normal distribution

Plotted Points

Each point on an $\bar{X}$ chart indicates the value of a subgroup mean ($\bar{X}_i$). For example, if the tenth subgroup contains the values 12, 15, 19, 16, and 14, the value plotted for this subgroup is

$$\bar{X}_{10} = \frac{12 + 15 + 19 + 16 + 14}{5} = 15.2$$

Central Line

By default, the central line on an $\bar{X}$ chart indicates an estimate for μ, which is computed as

$$\hat{\mu} = \overline{\overline{X}} = \frac{n_1\bar{X}_1 + \cdots + n_N\bar{X}_N}{n_1 + \cdots + n_N}$$

If you specify a known value (μ_0) for μ, the central line indicates the value of μ_0.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of $\bar{X}_i$ above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that $\bar{X}_i$ exceeds the limits

The following table provides the formulas for the limits:

Table 49.23. Limits for $\bar{X}$ Charts

Control Limits
LCL = lower limit = $\overline{\overline{X}} - k\hat{\sigma}/\sqrt{n_i}$ UCL = upper limit = $\overline{\overline{X}} + k\hat{\sigma}/\sqrt{n_i}$

Probability Limits
LCL = lower limit = $\overline{\overline{X}} - z_{\alpha/2}(\hat{\sigma}/\sqrt{n_i})$ UCL = upper limit = $\overline{\overline{X}} + z_{\alpha/2}(\hat{\sigma}/\sqrt{n_i})$

Note that the limits vary with n_i. If standard values μ_0 and σ_0 are available for μ and σ, respectively, replace $\overline{\overline{X}}$ with μ_0 and $\hat{\sigma}$ with σ_0 in Table 49.23.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.

- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify μ_0 with the MU0= option or with the variable _MEAN_ in a LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables can be saved:

Table 49.24. OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
_CP_	capability index C_p
CPK	capability index C_{pk}
CPL	capability index CPL
CPM	capability index C_{pm}
CPU	capability index CPU
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLR	lower control limit for subgroup range
LCLS	lower control limit for subgroup standard deviation
LCLX	lower control limit for subgroup mean
LIMITN	sample size associated with the control limits
LSL	lower specification limit
MEAN	process mean ($\overline{\overline{X}}$ or μ_0)
_R_	value of central line on R chart
_S_	value of central line on s chart
SIGMAS	multiple (k) of standard error of $\bar{X}_i$
STDDEV	process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*subgroup-variable* specified in the XCHART statement
TARGET	target value
TYPE	type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLR	upper control limit for subgroup range
UCLS	upper control limit for subgroup standard deviation
UCLX	upper control limit for subgroup mean
USL	upper specification limit
VAR	*process* specified in the XCHART statement

Notes:

1. The variables _LCLS_, _S_, and _UCLS_ are included if you specify the STDDEVIATIONS option; otherwise, the variables _LCLR_, _R_, and

UCLR are included. These variables are not used to create $\bar{X}$ charts, but they enable the OUTLIMITS= data set to be used as a LIMITS= data set with the BOXCHART, MRCHART, RCHART, SCHART, XRCHART, and XSCHART statements.

2. If the control limits vary with subgroup sample size, the special missing value *V* is assigned to the variables _LIMITN_, _LCLX_, _UCLX_, _LCLR_, _R_, _UCLR_, _LCLS_, _S_, and _UCLS_.
3. If the limits are defined in terms of a multiple k of the standard error of $\bar{X}_i$, the value of _ALPHA_ is computed as $\alpha = 2(1 - \Phi(k))$, where $\Phi(\cdot)$ is the standard normal distribution function.
4. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = \Phi^{-1}(1 - \alpha/2)$, where Φ^{-1} is the inverse standard normal distribution function.
5. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you provide a target value with the TARGET= option. See "Capability Indices" on page 1774 for computational details.
6. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the XCHART statement. For an example, see "Saving Control Limits" on page 1699.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables can be saved:

- the *subgroup-variable*
- a subgroup mean variable named by *process* suffixed with *X*
- a subgroup sample size variable named by *process* suffixed with *N*
- a subgroup range variable named by *process* suffixed with *R*
- a subgroup standard deviation variable named by *process* suffixed with *S*

A subgroup standard deviation variable is included if you specify the STDDEVIATIONS option; otherwise, a subgroup range variable is included.

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the XCHART statement. For example, consider the following statements:

```
proc shewhart data=steel;
   xchart (width diameter)*lot / outhistory=summary;
run;
```

The data set SUMMARY contains variables named LOT, WIDTHX, WIDTHR, WIDTHN, DIAMTERX, DIAMTERR, and DIAMTERN. The variables WIDTHR and DIAMTERR are included, since the STDDEVIATIONS option is not specified. If you specified the STDDEVIATIONS option, the data set SUMMARY would contain the variables WIDTHS and DIAMTERS rather than WIDTHR and DIAMTERR.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 1697.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables can be saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on $\bar{X}$ chart
LCLX	lower control limit for mean
LIMITN	nominal sample size associated with the control limits
MEAN	process mean
SIGMAS	multiple (k) of the standard error associated with control limits
subgroup	values of the subgroup variable
SUBN	subgroup sample size
SUBX	subgroup mean
TESTS	tests for special causes signaled on $\bar{X}$ chart
UCLX	upper control limit for mean
VAR	*process* specified in the XCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)
- _TREND_ (if the TRENDVAR= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved, depending on how the control limits are defined (with the ALPHA= or SIGMAS= option, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The k^{th} character of a value of _TESTS_ is k if Test k is positive at that subgroup. For example, if you request all eight tests and Tests 2 and 8 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, an 8 for the eighth character, and blanks for the other six characters.
3. The variables _EXLIM_ and _TESTS_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1699.

ODS Tables

The following table summarizes the ODS tables that you can request with the XCHART statement.

Table 49.25. ODS Tables Produced with the XCHART Statement

Table Name	Description	Options
XCHART	$\bar{X}$ chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the XCHART statement must be a SAS variable in the DATA= data set. This variable provides measurements that must be grouped into subgroup samples indexed by the *subgroup-variable*. The *subgroup-variable*, which is specified in the XCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. If the i^{th} subgroup contains n_i items, there should be n_i consecutive observations for which the value of the *subgroup-variable* is the index of the i^{th} subgroup. For example, if each subgroup contains five items and there are 30 subgroup samples, the DATA= data set should contain 150 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 1936).

For an example of a DATA= data set, see "Creating Charts for Means from Raw Data" on page 1692.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set CONLIMS:*

```
proc shewhart data=info limits=conlims;
   xchart weight*batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set; see Table 49.24 on page 1716. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLX_, _MEAN_, and _UCLX_, which specify the control limits directly
- the variables _MEAN_ and _STDDEV_, which are used to calculate the control limits according to the equations in Table 49.23 on page 1715

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.

*In Release 6.09 and in earlier releases, it is necessary to specify the READLIMITS option.

- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are ESTIMATE, STANDARD, STDMU, and STDSIGMA.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1701.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART, CUSUM, or MACONTROL procedures or to read output data sets created with SAS summarization procedures, such as PROC MEANS.

A HISTORY= data set used with the XCHART statement must contain the following:

- the *subgroup-variable*
- a subgroup mean variable for each *process*
- a subgroup sample size variable for each *process*
- either a subgroup range variable or subgroup standard deviation variable for each *process*

If you specify the STDDEVIATIONS option, the subgroup standard deviation variable must be included; otherwise, the subgroup range variable must be included.

The names of the subgroup mean, subgroup range or subgroup standard deviation, and subgroup sample size variables must be the *process* name concatenated with the suffix characters X, R or S, and N, respectively.

For example, consider the following statements:

```
proc shewhart history=summary;
   xchart (weight yldstren)*batch;
run;
```

The data set SUMMARY must include the variables BATCH, WEIGHTX, WEIGHTR, WEIGHTN, YLDSRENX, YLDSRENR, and YLDSRENN. If the STDDEVIATIONS option were specified in the preceding XCHART statement, it would be necessary for SUMMARY to include the variables BATCH, WEIGHTX, WEIGHTS, WEIGHTN, YLDSRENX, YLDSRENS, and YLDSRENN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 1936 for an example).

For an example of a HISTORY= data set, see "Creating Charts for Means from Subgroup Summary Data" on page 1694.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in Chapter 56, "Specialized Control Charts."

The following table lists the variables required in a TABLE= data set used with the XCHART statement:

Table 49.26. Variables Required in a TABLE= Data Set

Variable	Description
LCLX	lower control limit for mean
LIMITN	nominal sample size associated with the control limits
MEAN	process mean
subgroup-variable	values of the *subgroup-variable*
SUBN	subgroup sample size
SUBX	subgroup mean
UCLX	upper control limit for mean

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.

- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1699.

Methods for Estimating the Standard Deviation

When control limits are computed from the input data, three methods (referred to as default, MVLUE, and RMSDF) are available for estimating the process standard deviation σ. The method depends on whether you specify the STDDEVIATIONS option. If you specify this option, σ is estimated using subgroup standard deviations; otherwise, σ is estimated using subgroup ranges.

For an illustration of the methods, see Example 49.2 on page 1728.

Default Method Based on Subgroup Ranges

If you do not specify the STDDEVIATIONS option, the default estimate for σ is

$$\hat{\sigma} = \frac{R_1/d_2(n_1) + \cdots + R_N/d_2(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, and R_i is the sample range of the observations $x_{i1}, \ldots, x_{in_i}$ in the i^{th} subgroup.

$$R_i = \max_{1 \leq j \leq n_i}(x_{ij}) - \min_{1 \leq j \leq n_i}(x_{ij})$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$. The unbiasing factor $d_2(n_i)$ is defined so that, if the observations are normally distributed, the expected value of R_i is $d_2(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the *ASTM Manual on Presentation of Data and Control Chart Analysis* (1976).

Default Method Based on Subgroup Standard Deviations

If you specify the STDDEVIATIONS option, the default estimate for σ is

$$\hat{\sigma} = \frac{s_1/c_4(n_1) + \cdots + s_N/c_4(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, s_i is the sample standard deviation of the i^{th} subgroup

$$s_i = \sqrt{\frac{1}{n_i - 1}\sum_{j=1}^{n_i}(x_{ij} - \bar{X}_i)^2}$$

and

$$c_4(n_i) = \frac{\Gamma(n_i/2)\sqrt{2/(n_i - 1)}}{\Gamma((n_i - 1)/2)}$$

Here $\Gamma(\cdot)$ denotes the gamma function, and $\bar{X}_i$ denotes the i^{th} subgroup mean. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$. If the observations are normally distributed, the expected value of s_i is $c_4(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the *ASTM Manual on Presentation of Data and Control Chart Analysis* (1976).

MVLUE Method Based on Subgroup Ranges

If you do not specify the STDDEVIATIONS option and you specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). The MVLUE is a weighted average of N unbiased estimates of σ of the form $R_i/d_2(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{f_1 R_1/d_2(n_1) + \cdots + f_N R_N/d_2(n_N)}{f_1 + \cdots + f_N}$$

where

$$f_i = \frac{[d_2(n_i)]^2}{[d_3(n_i)]^2}$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The unbiasing factor $d_3(n_i)$ is defined so that, if the observations are normally distributed, the expected value of σ_{R_i} is $d_3(n_i)\sigma$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

MVLUE Method Based on Subgroup Standard Deviations

If you specify the STDDEVIATIONS option and SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). This estimate is a weighted average of N unbiased estimates of σ of the form $s_i/c_4(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{h_1 s_1/c_4(n_1) + \cdots + h_N s_N/c_4(n_N)}{h_1 + \cdots + h_N}$$

where

$$h_i = \frac{[c_4(n_i)]^2}{1 - [c_4(n_i)]^2}$$

A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

RMSDF Method Based on Subgroup Standard Deviations

If you specify the STDDEVIATIONS option and SMETHOD=RMSDF, a weighted root-mean-square estimate is computed for σ.

$$\hat{\sigma} = \frac{\sqrt{(n_1 - 1)s_1^2 + \cdots + (n_N - 1)s_N^2}}{c_4(n)\sqrt{n_1 + \cdots + n_N - N}}$$

The weights are the degrees of freedom $n_i - 1$. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$.

If the unknown standard deviation σ is constant across subgroups, the root-mean-square estimate is more efficient than the minimum variance linear unbiased estimate. However, in process control applications, it is generally not assumed that σ is constant, and if σ varies across subgroups, the root-mean-square estimate tends to be more inflated than the MVLUE.

Default Method Based on Individual Measurements

When each subgroup sample contains a single observation ($n_i \equiv 1$), the process standard deviation σ is estimated as $\hat{\sigma} = \bar{R}/d_2(2)$, where $\bar{R}$ is the average of the moving ranges of consecutive measurements taken in pairs. This is the method used to estimate σ for individual measurements and moving range charts. See page 1379 in Chapter 41, "IRCHART Statement."

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical	DATA=	*process*
Vertical	HISTORY=	subgroup mean variable
Vertical	TABLE=	_SUBX_

If you specify the TRENDVAR= option, you can provide distinct labels for the vertical axes of the $\bar{X}$ and trend charts by breaking the vertical axis into two parts with a split character. Specify the split character with the SPLIT= option. The first part labels the vertical axis of the $\bar{X}$ chart, and the second part labels the vertical axis of the trend chart.

For example, the following sets of statements specify the label *Residual Mean* for the vertical axis of the $\bar{X}$ chart and the label *Fitted Mean* for the vertical axis of the trend chart:

```
proc shewhart data=toolwear;
   xchart diameter*hour / split    = '/'
                          trendvar = fitted ;
   label diameter = 'Residual Mean/Fitted Mean';
run;

proc shewhart history=regdata;
   xchart diameter*hour / split    = '/'
                          trendvar = fitted;
   label diamterx = 'Residual Mean/Fitted Mean';
run;
```

In this example, the label assignments are in effect only for the duration of the procedure step, and they temporarily override any permanent labels associated with the variables.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples

This section provides advanced examples of the XCHART statement.

Example 49.1. Applying Tests for Special Causes

See SHWTEST in the SAS/QC Sample Library

This example illustrates how you can apply tests for special causes to make $\bar{X}$ charts more sensitive to special causes of variation.

The following statements create an $\bar{X}$ chart for the gap width measurements in the data set PARTS on page 1694 and tabulate the results:

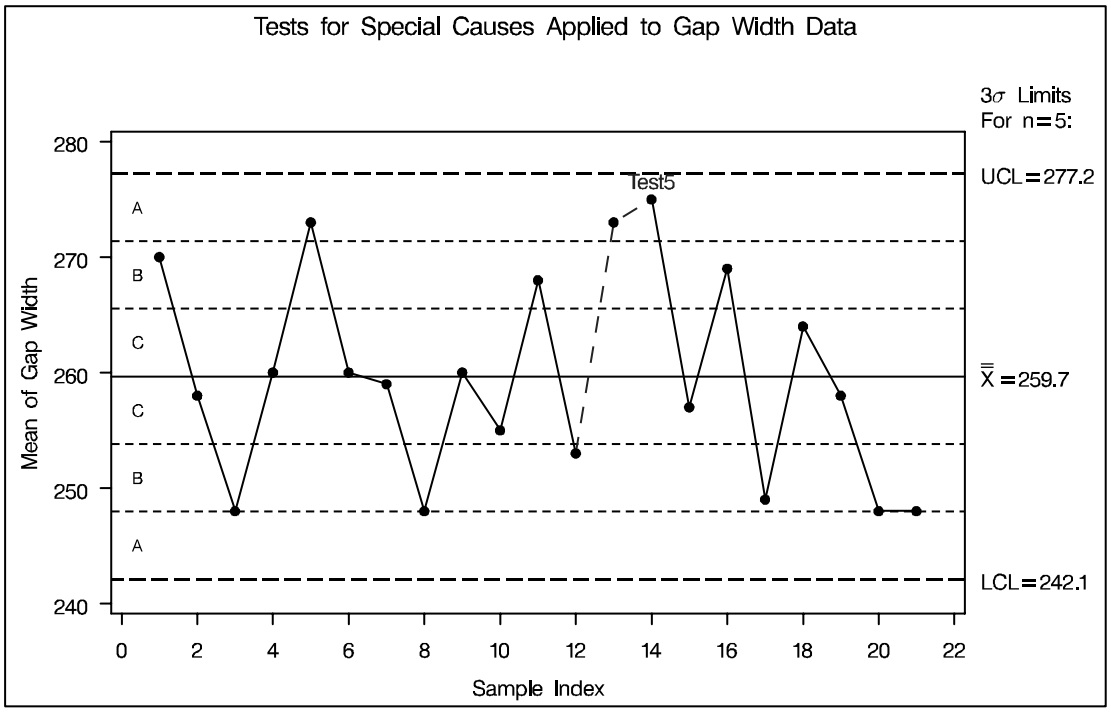
```
title 'Tests for Special Causes Applied to Gap Width Data';
proc shewhart history=parts;
   xchart partgap*sample/ tests   = 1 to 5
                          ltests  = 20
                          tabletests
                          nolegend
                          tablecentral
                          tablelegend
                          zonelabels;
run;
```

The $\bar{X}$ chart is shown in Output 49.1.1 and the printed output is shown in Output 49.1.2. The TESTS= requests Tests 1, 2, 3, 4, and 5, which are described in Chapter 55, "Tests for Special Causes." The TABLECENTRAL option requests a table of the subgroup means, control limits, and central line. The TABLETESTS option adds a column indicating which subgroups tested positive for special causes, and the TABLELEGEND option adds a legend describing the tests that were signaled.

The ZONELABELS option displays zone lines and zone labels on the chart. The zones are used to define the tests. The LTESTS= option specifies the line type used to connect the points in test patterns that were signaled. The NOLEGEND option suppresses the subgroup sample size legend that is displayed by default in the lower left corner of the chart.

Output 49.1.1 and Output 49.1.2 indicate that Test 5 was positive at sample 14, signaling a possible shift in the mean of the process.

Output 49.1.1. Tests for Special Causes Displayed on an $\bar{X}$ Chart



Output 49.1.2. Tabular Form of $\bar{X}$ Chart

Tests for Special Causes Applied to Gap Width Data

Means Chart Summary for partgap

sample	Subgroup Sample Size	3 Sigma Lower Limit	Subgroup Mean	Average Mean	3 Sigma Upper Limit	Special Tests Signaled
1	5	242.08741	270.00000	259.66667	277.24592	
2	5	242.08741	258.00000	259.66667	277.24592	
3	5	242.08741	248.00000	259.66667	277.24592	
4	5	242.08741	260.00000	259.66667	277.24592	
5	5	242.08741	273.00000	259.66667	277.24592	
6	5	242.08741	260.00000	259.66667	277.24592	
7	5	242.08741	259.00000	259.66667	277.24592	
8	5	242.08741	248.00000	259.66667	277.24592	
9	5	242.08741	260.00000	259.66667	277.24592	
10	5	242.08741	255.00000	259.66667	277.24592	
11	5	242.08741	268.00000	259.66667	277.24592	
12	5	242.08741	253.00000	259.66667	277.24592	
13	5	242.08741	273.00000	259.66667	277.24592	
14	5	242.08741	275.00000	259.66667	277.24592	5
15	5	242.08741	257.00000	259.66667	277.24592	
16	5	242.08741	269.00000	259.66667	277.24592	
17	5	242.08741	249.00000	259.66667	277.24592	
18	5	242.08741	264.00000	259.66667	277.24592	
19	5	242.08741	258.00000	259.66667	277.24592	
20	5	242.08741	248.00000	259.66667	277.24592	
21	5	242.08741	248.00000	259.66667	277.24592	

Test Descriptions

Test 5 Two out of three points in a row in Zone A or beyond

Example 49.2. Estimating the Process Standard Deviation

See SHWXEX2 in the SAS/QC Sample Library

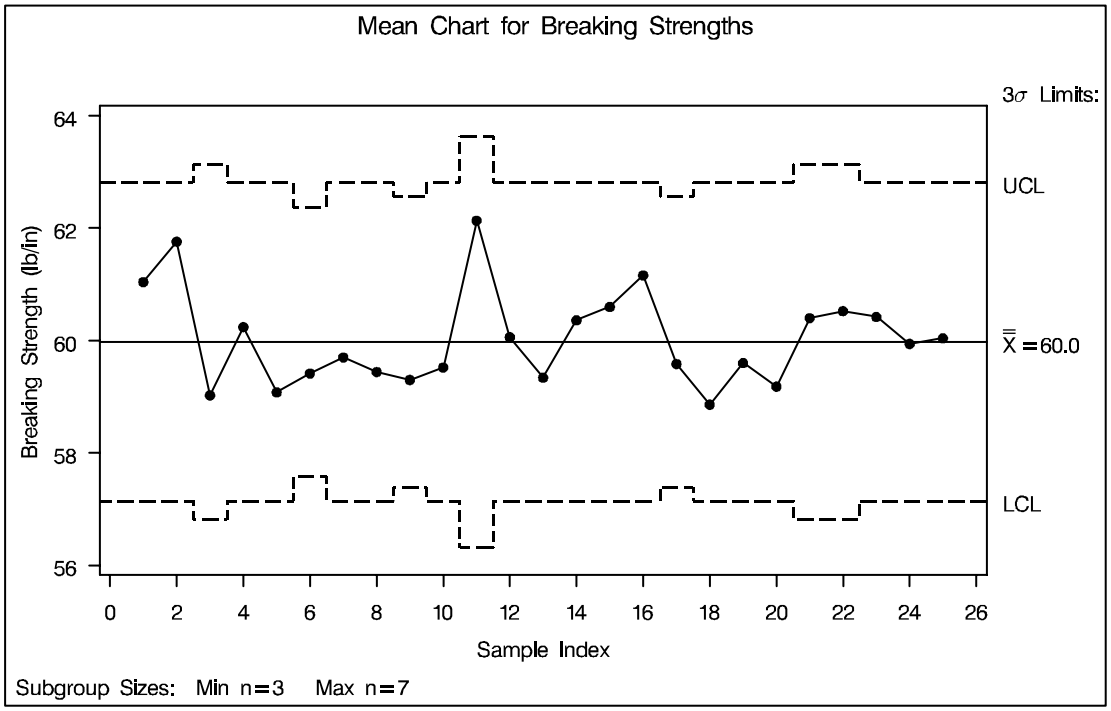
The following data set (WIRE) contains breaking strength measurements recorded in pounds per inch for 25 samples from a metal wire manufacturing process. The subgroup sample sizes vary between 3 and 7.

```
data wire;
   input sample size @;
   do i=1 to size;
      input brstr @@;
      output;
      end;
   drop i size;
   label brstr  ='Breaking Strength (lb/in)'
         sample ='Sample Index';
   datalines;
 1  5 60.6 62.3 62.0 60.4 59.9
 2  5 61.9 62.1 60.6 58.9 65.3
 3  4 57.8 60.5 60.1 57.7
 4  5 56.8 62.5 60.1 62.9 58.9
 5  5 63.0 60.7 57.2 61.0 53.5
 6  7 58.7 60.1 59.7 60.1 59.1 57.3 60.9
 7  5 59.3 61.7 59.1 58.1 60.3
 8  5 61.3 58.5 57.8 61.0 58.6
 9  6 59.5 58.3 57.5 59.4 61.5 59.6
10  5 61.7 60.7 57.2 56.5 61.5
11  3 63.9 61.6 60.9
12  5 58.7 61.4 62.4 57.3 60.5
13  5 56.8 58.5 55.7 63.0 62.7
14  5 62.1 60.6 62.1 58.7 58.3
15  5 59.1 60.4 60.4 59.0 64.1
16  5 59.9 58.8 59.2 63.0 64.9
17  6 58.8 62.4 59.4 57.1 61.2 58.6
18  5 60.3 58.7 60.5 58.6 56.2
19  5 59.2 59.8 59.7 59.3 60.0
20  5 62.3 56.0 57.0 61.8 58.8
21  4 60.5 62.0 61.4 57.7
22  4 59.3 62.4 60.4 60.0
23  5 62.4 61.3 60.5 57.7 60.2
24  5 61.2 55.5 60.2 60.4 62.4
25  5 59.0 66.1 57.7 58.5 58.9
;
run;
```

The following statements request an $\bar{X}$ chart, shown in Output 49.2.1, for the breaking strength measurements:

```
title 'Mean Chart for Breaking Strengths';
proc shewhart data=wire;
   xchart brstr*sample;
run;
```

Output 49.2.1. $\bar{X}$ Chart with Varying Subgroup Sample Sizes



Note that the control limits vary with the subgroup sample size. The sample size legend in the lower left corner displays the minimum and maximum subgroup sample sizes.

By default, the control limits are 3σ limits estimated from the data. You can use the STDDEVIATIONS option and the SMETHOD= option to specify how the estimate of the process standard deviation σ is to be computed, as illustrated by the following statements:

```
proc shewhart data=wire;
   xchart brstr*sample / outlimits=wirelim1
                         outindex ='Default-Ranges'
                         nochart;
   xchart brstr*sample / outlimits=wirelim2
                         stddeviations
                         outindex ='Default-Stds'
                         nochart;
   xchart brstr*sample / outlimits=wirelim3
                         smethod  =mvlue
                         outindex ='MVLUE  -Ranges'
                         nochart;
   xchart brstr*sample / outlimits=wirelim4
                         stddeviations
                         smethod  =mvlue
                         outindex ='MVLUE  -Stds'
                         nochart;
```

```
   xchart brstr*sample / outlimits=wirelim5
                         stddeviations
                         smethod  =rmsdf
                         outindex ='RMSDF  -Stds'
                         nochart;
run;
```

The STDDEVIATIONS option specifies that the estimate is to be calculated from subgroup standard deviations rather than subgroup ranges, the default. The SMETHOD= option specifies the method for estimating σ. The default method estimates σ as an unweighted average of subgroup estimates of σ. Specifying SMETHOD=MVLUE requests a minimum variance linear unbiased estimate, and specifying SMETHOD=RMSDF requests a weighted root-mean-square estimate. For details, see "Methods for Estimating the Standard Deviation" on page 1723.

The variable _STDDEV_ in each OUTLIMITS= data set contains the estimate of σ. The OUTINDEX= option specifies the value of the variable _INDEX_ in the OUTLIMITS= data set and is used here to identify the estimation method.

The following statements merge the five OUTLIMITS= data sets into a single data set, which is listed in Output 49.2.2:

```
data wlimits;
   set wirelim1 wirelim2 wirelim3 wirelim4 wirelim5;
   keep _index_ _stddev_;
run;
```

Output 49.2.2. The Data Set WLIMITS

Estimates of the Process Standard Deviation

INDEX	_STDDEV_
Default-Ranges	2.11146
Default-Stds	2.15453
MVLUE -Ranges	2.11240
MVLUE -Stds	2.14790
RMSDF -Stds	2.17479

The $\bar{X}$ chart shown in Output 49.2.1 uses the default estimate listed first in Output 49.2.2 ($\sigma = 2.11146$). In this case, there is very little difference in the five estimates, since the sample sizes do not differ greatly. In general, the MVLUE's are recommended with large sample sizes ($n_i \geq 10$).

Example 49.3. Plotting OC Curves for Mean Charts

See SHWOC1 in the SAS/QC Sample Library

This example uses the GPLOT procedure and the DATA step function PROBNORM to plot operating characteristic (OC) curves for $\bar{X}$ charts with 3σ limits. An OC curve is plotted for each of the subgroup samples sizes 1, 2, 3, 4, and 16. Refer to page 226 in Montgomery (1996). Each curve plots the probability β of not detecting a shift of magnitude $\nu\sigma$ in the process mean as a function of ν. The value of β is computed using the following formula:

$$\begin{aligned}\beta &= P\{LCL \leq \bar{X}_i \leq UCL\} \\ &= \Phi(3 - \nu\sqrt{n}) - \Phi(-3 - \nu\sqrt{n})\end{aligned}$$

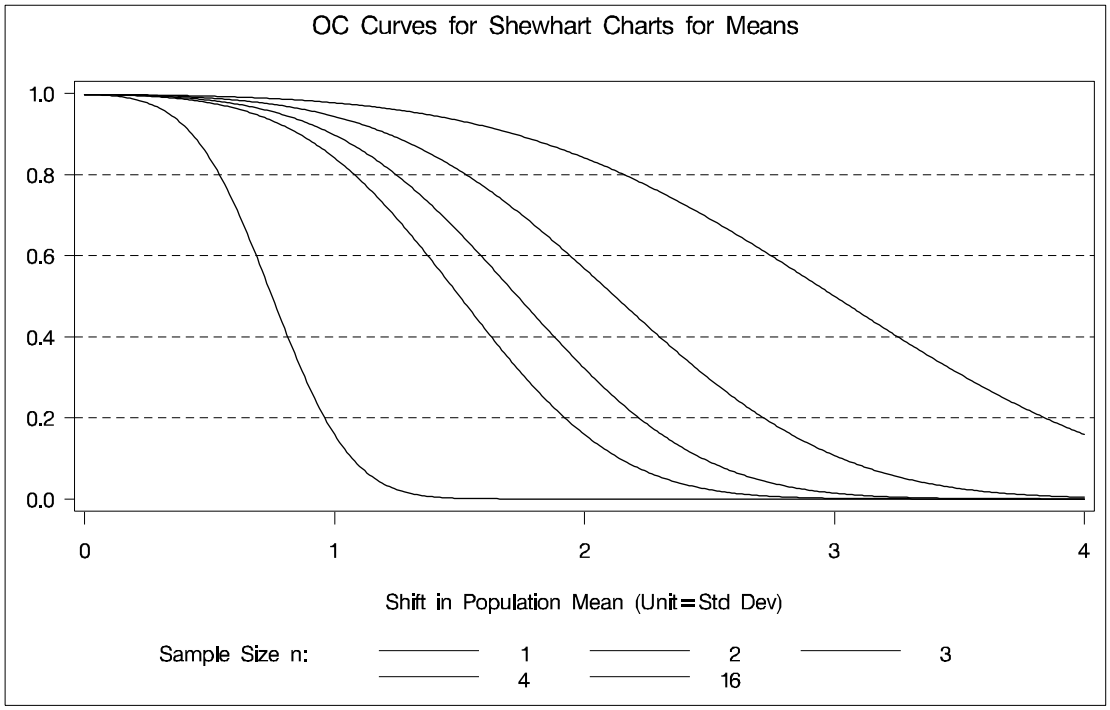
The following statements compute β (the variable BETA) as a function of ν (the variable NU). The variable NSAMPLE contains the sample size.

```
data oc;
   keep prob nsample t plot2;
   plot2=.;
   do nsample=1, 2, 3, 4, 16;
      do j=0 to 400;
         t=j/100;
         prob=probnorm( 3-t*sqrt(nsample)) -
              probnorm(-3-t*sqrt(nsample));
         output;
         end;
      end;
   label t    ='Shift in Population Mean (Unit=Std Dev)'
         prob='Probability of Not Detecting Shift';
run;
```

The following statements use the GPLOT procedure to display the OC curves shown in Output 49.3.1:

```
symbol1 v=none i=join w=2;
symbol2 v=none i=join w=2;
symbol3 v=none i=join w=2;
symbol4 v=none i=join w=2;
symbol5 v=none i=join w=2;
title1 'OC Curves for Shewhart Charts for Means';
proc gplot data=oc;
   plot prob*t=nsample /
      vminor = 0
      hminor = 0
      vref   = 0.2 0.4 0.6 0.8
      lvref  = 2
      vaxis  = axis1
      legend = legend1;

   axis1 label=(r=90 a=-90)
         order=(0.0 0.2 0.4 0.6 0.8 1.0);
   legend1 label=('Sample Size n:');
run;
```

Output 49.3.1. OC Curves for Different Subgroup Sample Sizes



Example 49.4. Computing Process Capability Indices

You can save process capability indices in an OUTLIMITS= data set if you provide specification limits with the LSL= and USL= options. This is illustrated by the following statements:

```
title 'Control Limits and Capability Indices';
proc shewhart data=partgaps;
   xchart partgap*sample / outlimits = gaplim2
                           usl       = 270
                           lsl       = 240
                           nochart;
run;
```

The data set GAPLIM2 is listed in Output 49.4.1.

Output 49.4.1. Data Set GAPLIM2 Containing Control Limit Information

Control Limits with Capability Indices for Gap Width Measurements

VAR	_SUBGRP_	_TYPE_	_LIMIT_	_ALPHA_	_SIGMAS_	_CLX_	_MEAN_	_CLX_	_CLR_
partgap	sample	ESTIMATE	5	.002699796	3	242.087	259.667	277.246	0

_R_	_UCLR_	_STDDEV_	_LSL_	_USL_	_CP_	_CPL_	_CPU_	_CPK_
30.4762	64.4419	13.1028	240	270	0.38160	0.50032	0.26288	0.26288

The variables _CP_, _CPL_, _CPU_, and _CPK_ contain the process capability indices. It is reasonable to compute capability indices in this case, since Figure 49.2 indicates that the process is in statistical control. For more information, see "OUTLIMITS= Data Set" on page 1716.

Chapter 50
XRCHART Statement

Chapter Contents

OVERVIEW 1737

GETTING STARTED 1738
- Creating Charts for Means and Ranges from Raw Data 1738
- Creating Charts for Means and Ranges from Summary Data 1741
- Saving Summary Statistics 1744
- Saving Control Limits 1745
- Reading Preestablished Control Limits 1748

SYNTAX 1750
- Summary of Options 1751

DETAILS 1762
- Constructing Charts for Means and Ranges 1762
- Output Data Sets 1764
- ODS Tables 1767
- Input Data Sets 1767
- Subgroup Variables 1771
- Methods for Estimating the Standard Deviation 1773
- Capability Indices 1774
- Axis Labels 1776
- Missing Values 1776

EXAMPLES 1777
- Example 50.1. Applying Tests for Special Causes 1777
- Example 50.2. Specifying Standard Values for the Process Mean and Standard Deviation 1780
- Example 50.3. Working with Unequal Subgroup Sample Sizes 1781

Chapter 50
XRCHART Statement

Overview

The XRCHART statement creates $\bar{X}$ and R charts for subgroup means and ranges, which are used to analyze the central tendency and variability of a process.

You can use options in the XRCHART statement to

- compute control limits from the data based on a multiple of the standard error of the plotted means and ranges or as probability limits
- tabulate subgroup sample sizes, subgroup means, subgroup ranges, control limits, and other information
- save control limits in an output data set
- save subgroup sample sizes, subgroup means, and subgroup ranges in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify a method for estimating the process standard deviation
- specify a known (standard) process mean and standard deviation for computing control limits
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the charts more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

Getting Started

This section introduces the XRCHART statement with simple examples that illustrate commonly used options. Complete syntax for the XRCHART statement is presented in the "Syntax" section on page 1750, and advanced examples are given in the "Examples" section on page 1777.

Creating Charts for Means and Ranges from Raw Data

See SHWXR1 in the SAS/QC Sample Library

In the manufacture of silicon wafers, batches of five wafers are sampled, and their diameters are measured in millimeters. The following statements create a SAS data set named WAFERS, which contains the measurements for 25 batches:

```
data wafers;
   input batch @;
   do i=1 to 5;
      input diamtr @;
      output;
      end;
   drop i;
datalines;
 1  35.00 34.99 34.99 34.98 35.00
 2  35.01 34.99 34.99 34.98 35.00
 3  34.99 35.00 35.00 35.00 35.00
 4  35.01 35.00 34.99 34.99 35.00
 5  35.00 34.99 34.98 34.99 35.00
 6  34.99 34.99 35.00 35.00 35.00
 7  35.01 34.98 35.00 35.00 34.99
 8  35.00 35.00 34.99 34.98 34.99
 9  34.99 34.98 34.98 35.01 35.00
10  34.99 35.00 35.01 34.99 35.01
11  35.01 35.00 35.00 34.98 34.99
12  34.99 34.99 35.00 34.98 35.01
13  35.01 34.99 34.98 34.99 34.99
14  35.00 35.00 34.99 35.01 34.99
15  34.98 34.99 34.99 34.98 35.00
16  34.99 35.00 35.00 35.01 35.00
17  34.98 34.98 34.99 34.99 34.98
18  35.01 35.02 35.00 34.98 35.00
19  34.99 34.98 35.00 34.99 34.98
20  34.99 35.00 35.00 34.99 34.99
21  35.00 34.99 34.99 34.98 35.00
22  35.00 35.00 35.01 35.00 35.00
23  35.02 35.00 34.98 35.02 35.00
24  35.00 35.00 34.99 35.01 34.98
25  34.99 34.99 34.99 35.00 35.00
;
run;
```

The following statements use the PRINT procedure to list the data set WAFERS. A portion of this listing is shown in Figure 50.1.

```
title 'The Data Set WAFERS';
proc print data=wafers noobs;
run;
```

The Data Set WAFERS

batch	diamtr
1	35.00
1	34.99
1	34.99
1	34.98
1	35.00
2	35.01
2	34.99
2	34.99
2	34.98
2	35.00
3	34.99
3	35.00
3	35.00
3	35.00
3	35.00
.	.
.	.
.	.

Figure 50.1. Partial Listing of the Data Set WAFERS

The data set WAFERS is said to be in "strung-out" form since each observation contains the batch number and diameter measurement for a single wafer. The first five observations contain the diameters for the first batch, the second five observations contain the diameters for the second batch, and so on. Because the variable BATCH classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable DIAMTR contains the wafer diameter measurements and is referred to as the *process variable* (or *process* for short).

You can use $\bar{X}$ and R charts to determine whether the manufacturing process is in control. The following statements create the $\bar{X}$ and R charts shown in Figure 50.2:

```
title 'Mean and Range Charts for Diameters';
proc shewhart data=wafers;
   xrchart diamtr*batch;
run;
```

This example illustrates the basic form of the XRCHART statement. After the keyword XRCHART, which specifies the type of control chart to display, you specify the *process* to analyze (in this case, DIAMTR) followed by an asterisk and the *subgroup-variable* (BATCH).

The input data set is specified with the DATA= option in the PROC SHEWHART statement.

If you use a graphics device, the SYMBOL statement specifies the symbol to plot the points. For more information on the SYMBOL statement, refer to *SAS/GRAPH Software: Reference*.

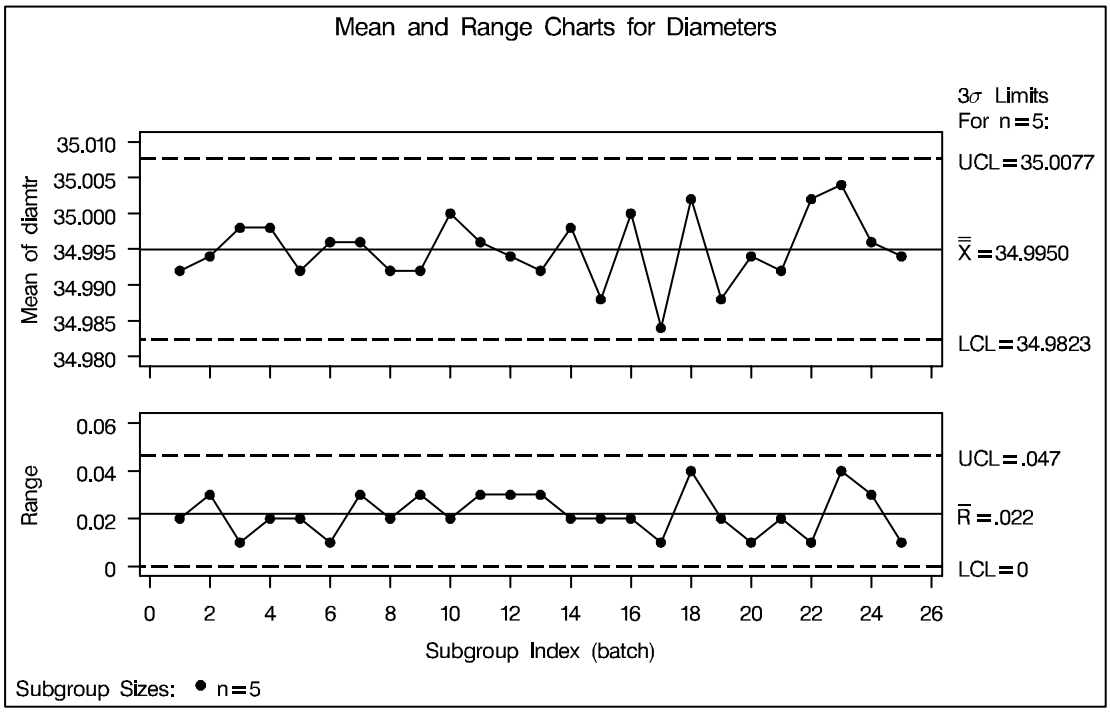


Figure 50.2. $\bar{X}$ and R Charts for Wafer Diameter Data

Each point on the $\bar{X}$ chart represents the average (mean) of the measurements for a particular batch. For instance, the mean plotted for the first batch is

$$\frac{35.00 + 34.99 + 34.99 + 34.98 + 35.00}{5} = 34.992$$

Each point on the R chart represents the range of the measurements for a particular batch. For instance, the range plotted for the first batch is $35.00 - 34.98 = 0.02$.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 50.23 on page 1763. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1748.

Since all the points lie within the control limits, it can be concluded that the process is in statistical control. For computational details, see "Constructing Charts for Means and Ranges" on page 1762. For more details on reading raw data, see "DATA= Data Set" on page 1767.

Creating Charts for Means and Ranges from Summary Data

See SHWXR1 in the SAS/QC Sample Library

The previous example illustrates how you can create $\bar{X}$ and R charts using raw data (process measurements). However, in many applications, the data are provided as subgroup means and ranges. This example illustrates how you can use the XRCHART statement with data of this type.

The following data set (WAFERSUM) provides the data from the preceding example in summarized form:

```
data wafersum;
   input batch diamtrx diamtrr;
   diamtrn = 5;
datalines;
 1  34.992  0.02
 2  34.994  0.03
 3  34.998  0.01
 4  34.998  0.02
 5  34.992  0.02
 6  34.996  0.01
 7  34.996  0.03
 8  34.992  0.02
 9  34.992  0.03
10  35.000  0.02
11  34.996  0.03
12  34.994  0.03
13  34.992  0.03
14  34.998  0.02
15  34.988  0.02
16  35.000  0.02
17  34.984  0.01
18  35.002  0.04
19  34.988  0.02
20  34.994  0.01
21  34.992  0.02
22  35.002  0.01
23  35.004  0.04
24  34.996  0.03
25  34.994  0.01
;
run;
```

A partial listing of the data set WAFERSUM is shown in Figure 50.3.

Summary Data Set for Wafer Diameters

batch	diamtrx	diamtrr	diamtrn
1	34.992	0.02	5
2	34.994	0.03	5
3	34.998	0.01	5
4	34.998	0.02	5
5	34.992	0.02	5
.	.	.	.
.	.	.	.
.	.	.	.

Figure 50.3. The Summary Data Set WAFERSUM

In this data set, there is exactly one observation for each subgroup (note that the subgroups are still indexed by BATCH). The variable DIAMTRX contains the subgroup means, the variable DIAMTRR contains the subgroup ranges, and the variable DIAMTRN contains the subgroup sample sizes (these are all equal to five).

You can read this data set by specifying it as a HISTORY= data set in the PROC SHEWHART statement, as follows:

```
title 'Mean and Range Charts for Diameters';
proc shewhart history=wafersum lineprinter;
   xrchart diamtr*batch='*';
run;
```

The charts are shown in Figure 50.4. Since the LINEPRINTER option is specified in the PROC SHEWHART statement, line printer output is produced. * The asterisk (*) specified in single quotes after the *subgroup-variable* indicates the character used to plot the points. This character must follow an equal sign.

Note that DIAMTR is *not* the name of a SAS variable in the data set WAFERSUM but is, instead, the common prefix for the names of the three SAS variables DIAMTRX, DIAMTRR, and DIAMTRN. The suffix characters *X*, *R*, and *N* indicate *mean*, *range*, and *sample size*, respectively. Thus, you can specify three subgroup summary variables in the HISTORY= data set with a single name (DIAMTR), which is referred to as the *process*. The name BATCH specified after the asterisk is the name of the *subgroup-variable*.

*In Release 6.12 and previous releases of SAS/QC software, the keyword GRAPHICS was required in the PROC SHEWHART statement to specify that the chart be created with a graphics device. In Version 7, you can specify the LINEPRINTER option to request line printer plots.

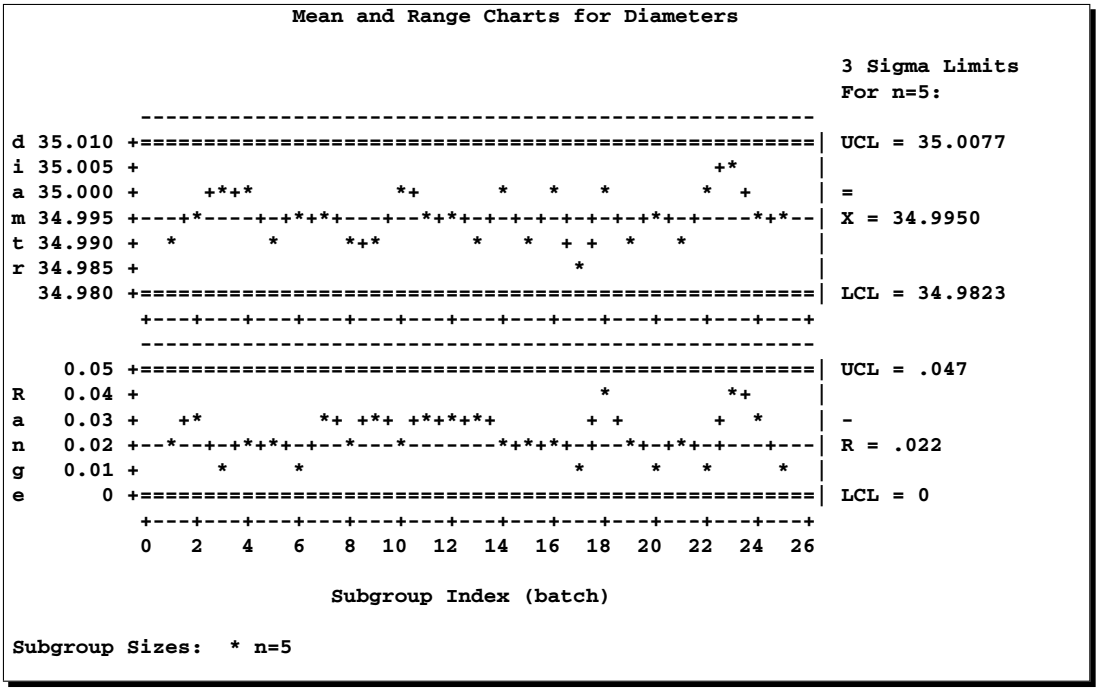


Figure 50.4. $\bar{X}$ and R Charts from Summary Data

In general, a HISTORY= input data set used with the XRCHART statement must contain the following variables:

- subgroup variable
- subgroup mean variable
- subgroup range variable
- subgroup sample size variable

Furthermore, the names of the subgroup mean, range, and sample size variables must begin with the *process* name specified in the XRCHART statement and end with the special suffix characters *X*, *R*, and *N*, respectively. If the names do not follow this convention, you can use the RENAME option to rename the variables for the duration of the SHEWHART procedure step. Suppose that, instead of the variables DIAMTRX, DIAMTRR, and DIAMTRN, the data set WAFERSUM contained summary variables named MEANS, RANGES, and SIZES. The following statements would temporarily rename MEANS, RANGES, and SIZES to DIAMTRX, DIAMTRR, and DIAMTRN, respectively:

```
proc shewhart
   history=wafersum (rename=(means  = diamtrx
                             ranges = diamtrr
                             sizes  = diamtrn ));
   xrchart diamtr*batch='*';
run;
```

In summary, the interpretation of *process* depends on the input data set:

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names of the variables containing the summary statistics.

For more information, see "HISTORY= Data Set" on page 1768.

Saving Summary Statistics

See SHWXR1 in the SAS/QC Sample Library

In this example, the XRCHART statement is used to create a summary data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read measurements from the data set WAFERS and create a summary data set named WAFRHIST:

```
proc shewhart data=wafers;
   xrchart diamtr*batch / outhistory = wafrhist
                          nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the charts. Options such as OUTHISTORY= and NOCHART are specified after the slash (/) in the XRCHART statement. A complete list of options is presented in the "Syntax" section on page 1750.

Figure 50.5 contains a partial listing of WAFRHIST.

Summary Data Set for Wafer Diameters

batch	diamtr X	diamtr R	diamtr N
1	34.992	0.02	5
2	34.994	0.03	5
3	34.998	0.01	5
4	34.998	0.02	5
5	34.992	0.02	5
.	.	.	.
.	.	.	.
.	.	.	.

Figure 50.5. The Summary Data Set WAFRHIST

There are four variables in the data set WAFRHIST.

- BATCH contains the subgroup index.
- DIAMTRX contains the subgroup means.
- DIAMTRR contains the subgroup ranges.
- DIAMTRN contains the subgroup sample sizes.

Note that the summary statistic variables are named by adding the suffix characters *X*, *R*, and *N* to the *process* DIAMTR specified in the XRCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 1765.

Saving Control Limits

See SHWXR1 in the SAS/QC Sample Library

You can save the control limits for $\bar{X}$ and R charts in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1748) or modify the limits with a DATA step program.

The following statements read measurements from the data set WAFERS (see page 1738) and save the control limits displayed in Figure 50.2 in WAFERLIM:

```
proc shewhart data=wafers;
   xrchart diamtr*batch / outlimits = waferlim
                          nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the charts. The data set WAFERLIM is listed in Figure 50.6.

Control Limits for Wafer Diameters

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_
diamtr	batch	ESTIMATE	5	.002699796	3	34.9823

MEAN	_UCLX_	_LCLR_	_R_	_UCLR_	_STDDEV_
34.9950	35.0077	0	0.022	0.046519	.009458586

Figure 50.6. The Data Set WAFERLIM Containing Control Limit Information

The data set WAFERLIM contains one observation with the limits for *process* DIAMTR. The variables _LCLX_ and _UCLX_ contain the lower and upper control limits for the $\bar{X}$ chart. The variables _LCLR_ and _UCLR_ contain the lower and upper control limits for the R chart. The variable _MEAN_ contains the central line for the $\bar{X}$ chart, and the variable _R_ contains the central line for the R chart. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values.

You can save process capability indices in an OUTLIMITS= data set if you provide specification limits with the LSL= and USL= options. This is illustrated by the following statements:

```
proc shewhart data=wafers;
   xrchart diamtr*batch / outlimits = wafrlim2
                          usl       = 35.03
                          lsl       = 34.97
                          nochart;
run;
```

The data set WAFRLIM2 is listed in Figure 50.7.

Control Limits and Capability Indices

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_	_MEAN_	_UCLX_	_LCLR_	_R_
diamtr	batch	ESTIMATE	5	.002699796	3	34.9823	34.9950	35.0077	0	0.022

UCLR	_STDDEV_	_LSL_	_USL_	_CP_	_CPL_	_CPU_	_CPK_
0.046519	.009458586	34.97	35.03	1.05724	0.87962	1.23486	0.87962

Figure 50.7. The Data Set WAFRLIM2 Containing Process Capability Indices

The variables _CP_, _CPL_, _CPU_, and _CPK_ contain the process capability indices. It is reasonable to compute capability indices, since Figure 50.2 indicates that the wafer process is in statistical control. However, it is recommended that you also check for normality of the data. You can use the CAPABILITY procedure for this purpose.

For more information, see "OUTLIMITS= Data Set" on page 1764.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=wafers;
   xrchart diamtr*batch / outtable=wtable
                          nochart;
run;
```

The data set WTABLE is listed in Figure 50.8.

Summary Statistics and Control Limit Information

VAR	BATCH	_SIGMA_	_LIMITS_	_SUBX_	_LCLX_	_MEAN_	_UCLX_	_EXLIM_	_LCLR_	_SUBR_	_UCLR_	_EXLIM_
diamtr	1	3	5	5	34.9823	34.992	34.9950	35.0077	0	0.02	0.022	0.046519
diamtr	2	3	5	5	34.9823	34.994	34.9950	35.0077	0	0.03	0.022	0.046519
diamtr	3	3	5	5	34.9823	34.998	34.9950	35.0077	0	0.01	0.022	0.046519
diamtr	4	3	5	5	34.9823	34.998	34.9950	35.0077	0	0.02	0.022	0.046519
diamtr	5	3	5	5	34.9823	34.992	34.9950	35.0077	0	0.02	0.022	0.046519
diamtr	6	3	5	5	34.9823	34.996	34.9950	35.0077	0	0.01	0.022	0.046519
diamtr	7	3	5	5	34.9823	34.996	34.9950	35.0077	0	0.03	0.022	0.046519
diamtr	8	3	5	5	34.9823	34.992	34.9950	35.0077	0	0.02	0.022	0.046519
diamtr	9	3	5	5	34.9823	34.992	34.9950	35.0077	0	0.03	0.022	0.046519
diamtr	10	3	5	5	34.9823	35.000	34.9950	35.0077	0	0.02	0.022	0.046519
diamtr	11	3	5	5	34.9823	34.996	34.9950	35.0077	0	0.03	0.022	0.046519
diamtr	12	3	5	5	34.9823	34.994	34.9950	35.0077	0	0.03	0.022	0.046519
diamtr	13	3	5	5	34.9823	34.992	34.9950	35.0077	0	0.03	0.022	0.046519
diamtr	14	3	5	5	34.9823	34.998	34.9950	35.0077	0	0.02	0.022	0.046519
diamtr	15	3	5	5	34.9823	34.988	34.9950	35.0077	0	0.02	0.022	0.046519
diamtr	16	3	5	5	34.9823	35.000	34.9950	35.0077	0	0.02	0.022	0.046519
diamtr	17	3	5	5	34.9823	34.984	34.9950	35.0077	0	0.01	0.022	0.046519
diamtr	18	3	5	5	34.9823	35.002	34.9950	35.0077	0	0.04	0.022	0.046519
diamtr	19	3	5	5	34.9823	34.988	34.9950	35.0077	0	0.02	0.022	0.046519
diamtr	20	3	5	5	34.9823	34.994	34.9950	35.0077	0	0.01	0.022	0.046519
diamtr	21	3	5	5	34.9823	34.992	34.9950	35.0077	0	0.02	0.022	0.046519
diamtr	22	3	5	5	34.9823	35.002	34.9950	35.0077	0	0.01	0.022	0.046519
diamtr	23	3	5	5	34.9823	35.004	34.9950	35.0077	0	0.04	0.022	0.046519
diamtr	24	3	5	5	34.9823	34.996	34.9950	35.0077	0	0.03	0.022	0.046519
diamtr	25	3	5	5	34.9823	34.994	34.9950	35.0077	0	0.01	0.022	0.046519

Figure 50.8. The Data Set WTABLE

This data set contains one observation for each subgroup sample. The variables _SUBX_, _SUBR_, and _SUBN_ contain the subgroup means, subgroup ranges, and subgroup sample sizes. The variables _LCLX_ and _UCLX_ contain the lower and upper control limits for the $\bar{X}$ chart. The variables _LCLR_ and _UCLR_ contain the lower and upper control limits for the R chart. The variable _MEAN_ contains the central line of the $\bar{X}$ chart, and the variable _R_ contains the central line of the R chart. The variables _VAR_ and BATCH contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1766.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read WTABLE and display $\bar{X}$ and R charts identical to those in Figure 50.2:

```
title 'Mean and Range Charts for Diameters';
proc shewhart table=wtable;
   xrchart diamtr*batch;
run;
```

Because the SHEWHART procedure simply displays the information read from a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see Chapter 56, "Specialized Control Charts,").

For more information, see "TABLE= Data Set" on page 1769.

Reading Preestablished Control Limits

See SHWXR1 in the SAS/QC Sample Library

In the previous example, the OUTLIMITS= data set saved control limits computed from the measurements in WAFERS. This example shows how these limits can be applied to new data provided in the following data set:

```
data wafers2;
   input batch @;
   do i=1 to 5;
      input diamtr @;
      output;
      end;
   drop i;
   datalines;
26 34.99 34.99 35.00 34.99 35.00
27 34.99 35.01 34.98 34.98 34.97
28 35.00 34.99 34.99 34.99 35.01
29 34.98 34.96 34.98 34.98 34.99
30 34.98 35.00 34.98 34.98 34.99
31 35.00 35.00 34.99 35.01 35.01
32 35.00 34.99 34.98 34.98 35.00
33 34.98 35.00 34.99 35.00 35.01
34 35.00 34.97 35.00 34.99 35.01
35 34.99 34.99 34.98 34.99 34.98
36 35.01 34.98 34.99 34.99 35.00
37 35.01 34.99 34.97 34.98 35.00
38 34.98 34.99 35.00 34.98 35.00
39 34.99 34.99 34.99 34.99 35.01
40 34.99 35.01 35.00 35.01 34.99
41 34.99 35.00 34.99 34.98 34.99
42 35.00 34.99 34.98 34.99 35.00
43 34.99 34.98 34.98 34.99 34.99
44 35.00 35.00 34.98 35.00 34.99
45 34.99 34.99 35.00 34.99 34.99
;
run;
```

The following statements create $\bar{X}$ and R charts for the data in WAFERS2 using the control limits in WAFERLIM:

```
title 'Mean and Range Charts for Diameters';
proc shewhart data=wafers2 limits=waferlim;
   xrchart diamtr*batch;
run;
```

The charts are shown in Figure 50.9.

The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default,* this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name DIAMTR
- the value of _SUBGRP_ matches the *subgroup-variable* name BATCH

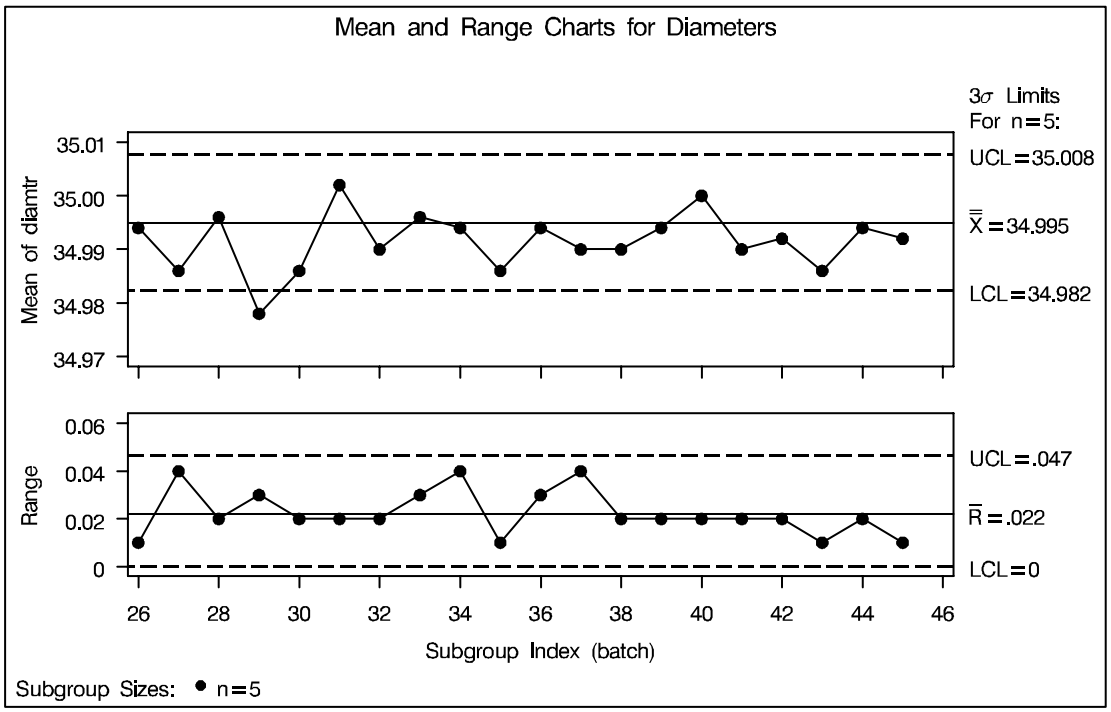


Figure 50.9. $\bar{X}$ and R Charts for Second Set of Wafer Data

Note that the mean diameter of the 29^{th} batch lies below the lower control limit in the $\bar{X}$ chart, signaling a special cause of variation.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1768 for details concerning the variables that you must provide.

*In Release 6.09 and in earlier releases, it is also necessary to specify the READLIMITS option to read control limits from a LIMITS= data set.

Syntax

The basic syntax for the XRCHART statement is as follows:

XRCHART *process***subgroup-variable* **;**

The general form of this syntax is as follows:

XRCHART (*processes***)****subgroup-variable* <**(***block-variables* **)** >
< =*symbol-variable* | =*'character'* > < **/** *options* >**;**

You can use any number of XRCHART statements in the SHEWHART procedure. The components of the XRCHART statement are described as follows.

process
processes
identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating Charts for Means and Ranges from Raw Data" on page 1738.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating Charts for Means and Ranges from Summary Data" on page 1741.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1745.

A *process* is required. If you specify more than one *process*, enclose the list in parentheses. For example, the following statements request distinct $\bar{X}$ and R charts for WEIGHT, LENGTH, and WIDTH:

```
proc shewhart data=measures;
   xrchart (weight length width)*day;
run;
```

subgroup-variable
is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding XRCHART statement, DAY is the subgroup variable. For details, see "Subgroup Variables" on page 1771.

block-variables
are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 1932 for an example.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the means and ranges.

- If you produce a chart on a line printer, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce a chart on a graphics device, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOLn statements. See "Displaying Stratification in Levels of a Classification Variable" on page 1931 for an example.

character
specifies a plotting character for charts produced on line printers. For example, the following statements create $\bar{X}$ and R charts using an asterisk (*) to plot the points:

```
proc shewhart data=values;
   xrchart weight*day='*';
run;
```

options
enhance the appearance of the charts, request additional analyses, save results in data sets, and so on. The "Summary of Options" section, which follows, lists all options by function. Chapter 53, "Dictionary of Options," describes each option in detail.

Summary of Options

The following tables list the XRCHART statement options by function. For complete descriptions, see Chapter 53, "Dictionary of Options."

Table 50.1. Tabulation Options

TABLE	creates a basic table of subgroup means, subgroup ranges, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with columns indicating which tests for special causes are positive

Note that specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.

Table 50.2. Options for Specifying Tests for Special Causes

NO3SIGMACHECK	enables tests to be applied with control limits other than 3σ limits
TESTS=*value-list*| *customized-pattern-list*	specifies tests for special causes for the $\bar{X}$ chart
TESTS2=*value-list*| *customized-pattern-list*	specifies tests for special causes for the R chart
TEST2RESET=*variable*	enables tests for special causes to be reset for the R chart
TEST2RUN=*n*	specifies length of pattern for Test 2
TEST3RUN=*n*	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL='*label*'| (*variable*)|*keyword*	provides labels for points where test is positive
TESTLABEL*n*='*label*'	specifies label for n^{th} test for special causes
TESTNMETHOD= STANDARDIZE	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
TESTRESET=*variable*	enables tests for special causes to be reset for the $\bar{X}$ chart
ZONELABELS	adds labels A, B, and C to zone lines for $\bar{X}$ chart
ZONE2LABELS	adds labels A, B, and C to zone lines for R chart
ZONES	adds lines to $\bar{X}$ chart delineating zones A, B, and C
ZONES2	adds lines to R chart delineating zones A, B, and C
ZONEVALPOS=*n*	specifies position of ZONEVALUES and ZONE2VALUES labels
ZONEVALUES	labels $\bar{X}$ chart zone lines with their values
ZONE2VALUES	labels R chart zone lines with their values

Table 50.3. Graphical Options for Displaying Tests for Special Causes

CTESTLABBOX=*color*	specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=*color*| *test-color-list*	specifies color for labels used to identify points where test is positive
CTESTSYMBOL=*color*	specifies color for symbol used to plot points where test is positive
CZONES=*color*	specifies color for lines and labels delineating zones A, B, and C
LTESTS=*linetype*	specifies type of line connecting points where test is positive
LZONES=*linetype*	specifies line type for lines delineating zones A, B, and C
TESTFONT=*font*	specifies software font for labels at points where test is positive
TESTHEIGHT=*value*	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=*symbol*	specifies plot symbol for points where test is positive
TESTSYMBOLHT=*value*	specifies symbol height for points where test is positive
WTESTS=*n*	specifies width of line connecting points where test is positive

Table 50.4. Line Printer Options for Displaying Tests for Special Causes

TESTCHAR='*character*'	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
ZONECHAR='*character*'	specifies character for lines that delineate zones for tests for special causes

Table 50.5. Reference Line Options

CHREF=*color*	specifies color for HREF= and HREF2= lines
CVREF=*color*	specifies color for VREF= and VREF2= lines
HREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on $\bar{X}$ chart
HREF2=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on R chart
HREFCHAR='*character*'	specifies line character for HREF= and HREF2= lines
HREFDATA= *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on $\bar{X}$ chart
HREF2DATA= *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on R chart
HREFLABELS= *'label1'...'labeln'*	specifies labels for HREF= lines
HREF2LABELS= *'label1'...'labeln'*	specifies labels for HREF2= lines
HREFLABPOS=*n*	specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=*linetype*	specifies line type for HREF= and HREF2= lines
LVREF=*linetype*	specifies line type for VREF= and VREF2= lines
NOBYREF	specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to vertical axis on $\bar{X}$ chart
VREF2=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to vertical axis on R chart
VREFCHAR='*character*'	specifies line character for VREF= and VREF2= lines
VREFLABELS= *'label1'...'labeln'*	specifies labels for VREF= lines
VREF2LABELS= *'label1'...'labeln'*	specifies labels for VREF2= lines
VREFLABPOS=*n*	specifies position of VREFLABELS= and VREF2LABELS= labels

Table 50.6. Block Variable Legend Options

BLOCKLABELPOS=*keyword*	specifies position of label for *block-variable* legend
BLOCKLABTYPE=*value*|*keyword*	specifies text size of *block-variable* legend
BLOCKPOS=*n*	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=*color-list*	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=*variable*|(*variables*)	specifies one or more variables whose values are colors for filling background of *block-variable* legend

Table 50.7. Axis and Axis Label Options

CAXIS=*color*	specifies color for axis lines and tick marks
CFRAME=*color*|(*color-list*)	specifies fill colors for frame for plot area
CTEXT=*color*	specifies color for tick mark values and axis labels
HAXIS=*values*|AXISn	specifies major tick mark values for horizontal axis
HEIGHT=*value*	specifies height of axis label and axis legend text
HMINOR=*n*	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=*value*	specifies length of offset at both ends of horizontal axis
INTSTART=*value*	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	suppresses vertical axis truncation at zero applied by default to R chart
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=*n*	specifies thinning factor for tick mark labels on horizontal axis
SPLIT='*character*'	specifies splitting character for axis labels
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=*values*|AXISn	specifies major tick mark values for vertical axis of $\bar{X}$ chart
VAXIS2=*values*|AXISn	specifies major tick mark values for vertical axis of R chart
VFORMAT=*format*	specifies format for primary vertical axis tick mark labels
VFORMAT2=*format*	specifies format for secondary vertical axis tick mark labels
VMINOR=n	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=*value*	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis for primary chart
VZERO2	forces origin to be included in vertical axis for secondary chart
WAXIS=*n*	specifies width of axis lines

Table 50.8. Specification Limit Options

CIINDICES (ALPHA=*value* TYPE=*keyword*)	specifies α value and type for computing capability index confidence limits
LSL=*value-list*	specifies list of lower specification limits
TARGET=*value-list*	specifies list of target values
USL=*value-list*	specifies list of upper specification limits

Table 50.9. Options for Specifying Control Limits

ALPHA=*value*	requests probability limits for control charts
LIMITN=*n*|VARYING	specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	computes control limits for each *process* from the data rather than from a LIMITS= data set (Release 6.10 and later releases)
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEXES=ALL| '*label1*'...'*labeln*'	reads multiple sets of control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (Release 6.09 and earlier releases)
SIGMAS=k	specifies width of control limits in terms of multiple k of standard error of plotted means and ranges

Table 50.10. Graphical Enhancement Options

ANNOTATE=*SAS-data-set*	specifies annotate data set that adds features to $\bar{X}$ chart
ANNOTATE2=*SAS-data-set*	specifies annotate data set that adds features to R chart
DESCRIPTION='*string*'	specifies string that appears in the description field of the PROC GREPLAY master menu for $\bar{X}$ chart
DESCRIPTION2='*string*'	specifies string that appears in the description field of the PROC GREPLAY master menu for R chart
FONT=*font*	specifies software font for labels and legends on charts
NAME='*string*'	specifies name that appears in the name field of the PROC GREPLAY master menu for $\bar{X}$ chart
NAME2='*string*'	specifies name that appears in the name field of the PROC GREPLAY master menu for R chart
PAGENUM='*string*'	specifies the form of the label used in pagination
PAGENUMPOS= *keyword*	specifies the position of the page number requested with the PAGENUM= option

Table 50.11. Options for Displaying Control Limits

CINFILL=*color*	specifies color for area inside control limits
CLIMITS=*color*	specifies color of control limits, central line, and related labels
LCLLABEL='*label*'	specifies label for lower control limit on $\bar{X}$ chart
LCLLABEL2='*label*'	specifies label for lower control limit on R chart
LIMLABSUBCHAR='*character*'	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=*linetype*	specifies line type for control limits
NDECIMAL=*n*	specifies number of digits to right of decimal place in default labels for control limits and central line on $\bar{X}$ chart
NDECIMAL2=*n*	specifies number of digits to right of decimal place in default labels for control limits and central line on R chart
NOCTL	suppresses display of central line on $\bar{X}$ chart
NOCTL2	suppresses display of central line on R chart
NOLCL	suppresses display of lower control limit on $\bar{X}$ chart
NOLCL2	suppresses display of lower control limit on R chart
NOLIMITLABEL	suppresses labels for control limits and central lines
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOLIMIT0	suppresses display of zero lower control limit on R chart
NOUCL	suppresses display of upper control limit on $\bar{X}$ chart
NOUCL2	suppresses display of upper control limit on R chart
RSYMBOL='*string*'|*keyword*	specifies label for central line on R chart
UCLLABEL='*string*'	specifies label for upper control limit on $\bar{X}$ chart
UCLLABEL2='*string*'	specifies label for upper control limit on R chart
WLIMITS=*n*	specifies width for control limits and central line
XSYMBOL='*string*'|*keyword*	specifies label for central line on $\bar{X}$ chart

Table 50.12. Grid Options

CGRID=*color*	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=*linetype*	specifies line type for grid requested with the ENDGRID option
LGRID=*linetype*	specifies line type for grid requested with the GRID option
WGRID=*n*	specifies width of grid lines

Table 50.13. Options for Plotting and Labeling Points

ALLLABEL=VALUE|(*variable*)	labels every point on $\bar{X}$ chart
ALLLABEL2=VALUE|(*variable*)	labels every point on R chart
CLABEL=*color*	specifies color for labels
CCONNECT=*color*	specifies color for line segments that connect points on chart
CFRAMELAB=*color*	specifies fill color for frame around labeled points
CNEEDLES=*color*	specifies color for needles that connect points to central line
CONNECTCHAR='*character*'	specifies character used to form line segments that connect points on chart
COUT=*color*	specifies color for portions of line segments that connect points outside control limits
COUTFILL=*color*	specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=*angle*	specifies angle at which labels are drawn
LABELFONT=*font*	specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=*value*	specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on chart
OUTLABEL=VALUE|(*variable*)	labels points outside control limits on $\bar{X}$ chart
OUTLABEL2=VALUE|(*variable*)	labels points outside control limits on R chart
SYMBOLCHARS='*characters*'	specifies characters indicating *symbol-variable*
SYMBOLLEGEND=NONE|*name*	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=*keyword*	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=*n*	specifies width of needles

Table 50.14. Clipping Options

CCLIP=*color*	specifies color for plot symbol for clipped points
CLIPCHAR='*character*'	specifies plot character for clipped points
CLIPFACTOR=*value*	determines extent to which extreme points are clipped
CLIPLEGEND='*string*'	specifies text for clipping legend
CLIPLEGPOS=*keyword*	specifies position of clipping legend
CLIPSUBCHAR='*character*'	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=*symbol*	specifies plot symbol for clipped points
CLIPSYMBOLHT=*value*	specifies symbol marker height for clipped points

Table 50.15. Phase Options

CPHASELEG=*color*	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE='*string*'	specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=*value*|*keyword*	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of charts
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES= ALL|'*label1*'...'*labeln*'	specifies *phases* to be read from an input data set

Table 50.16. Input Data Set Options

MISSBREAK	specifies that observations with missing values are not to be processed

Table 50.17. Output Data Set Options

OUTHISTORY=*SAS-data-set*	creates output data set containing subgroup summary statistics
OUTINDEX='*string*'	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=*SAS-data-set*	creates output data set containing control limits
OUTTABLE=*SAS-data-set*	creates output data set containing subgroup summary statistics and control limits

Table 50.18. Process Mean and Standard Deviation Options

MU0=*value*	specifies known (standard) value μ_0 for process mean μ
SIGMA0=*value*	specifies known (standard) value σ_0 for process standard deviation σ
SMETHOD=*keyword*	specifies method for estimating process standard deviation σ
TYPE=*keyword*	identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set

Table 50.19. Plot Layout Options

ALLN	plots means and ranges for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts for a *process* only when exceptions occur
INTERVAL=*keyword*	specifies natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=*n*	specifies maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of charts
NOCHART2	suppresses creation of R chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=*n*	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
SEPARATE	displays $\bar{X}$ and R charts on separate screens or pages
TOTPANELS=*n*	specifies number of pages or screens to be used to display chart
YPCT1=*value*	specifies length of vertical axis on $\bar{X}$ chart as a percentage of sum of lengths of vertical axes for $\bar{X}$ and R charts
ZEROSTD	displays $\bar{X}$ and R charts regardless of whether $\hat{\sigma} = 0$

Table 50.20. Options for Interactive Control Charts

HTML=(*variable*)	specifies a variable whose values are URLs to be associated with subgroups
HTML2=(*variable*)	specifies variable whose values are URLs to be associated with subgroups on secondary chart
HTML_LEGEND=(*variable*)	specifies a variable whose values are URLs to be associated with symbols in the symbol legend
WEBOUT=*SAS-data-set*	creates an OUTTABLE= data set with additional graphics coordinate data

Table 50.21. Star Options

CSTARCIRCLES=*color*	specifies color for STARCIRCLES= circles
CSTARFILL=*color*| *(variable)*	specifies color for filling stars
CSTAROUT=*color*	specifies outline color for stars exceeding inner or outer circles
CSTARS=*color*| *(variable)*	specifies color for outlines of stars
LSTARCIRCLES= *linetypes*	specifies line types for STARCIRCLES= circles
LSTARS=*linetype*|*(variable)*	specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=*value*	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=*value-list*	specifies reference circles for stars
STARINRADIUS=*value*	specifies inner radius of stars
STARLABEL=*keyword*	specifies vertices to be labeled
STARLEGEND=*keyword*	specifies style of legend for star vertices
STARLEGENDLAB=*'label'*	specifies label for STARLEGEND= legend
STAROUTRADIUS=*value*	specifies outer radius of stars
STARSPEC=*value*| *SAS-data-set*	specifies method used to standardize vertex variables
STARSTART=*value*	specifies angle for first vertex
STARTYPE=*keyword*	specifies graphical style of star
STARVERTICES= *variable*|*(variables)*	superimposes star at each point on $\bar{X}$ chart
WSTARCIRCLES=*n*	specifies width of STARCIRCLES= circles
WSTARS=*n*	specifies width of STARVERTICES= stars

Table 50.22. Overlay Options

CCOVERLAY=*color-list*	specifies colors for primary chart overlay line segments
CCOVERLAY2=*color-list*	specifies colors for secondary chart overlay line segments
COVERLAY=*color-list*	specifies colors for primary chart overlay plots
COVERLAY2=*color-list*	specifies colors for secondary chart overlay plots
COVERLAYCLIP=*color*	specifies color for clipped points on overlays
LOVERLAY=*linetypes*	specifies line types for primary chart overlay line segments
LOVERLAY2=*linetypes*	specifies line types for secondary chart overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=*variable-list*	specifies variables to overlay on primary chart
OVERLAY2=*variable-list*	specifies variables to overlay on secondary chart
OVERLAY2HTML=*variable-list*	specifies URLS to associate with secondary chart overlay points
OVERLAY2ID=*variable-list*	specifies labels for secondary chart overlay points
OVERLAY2SYM=*symbol-list*	specifies symbols for secondary chart overlays
OVERLAY2SYMHT=*value-list*	specifies symbol heights for secondary chart overlays
OVERLAYCLIPSYM=*symbol*	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=*value*	specifies symbol height for clipped points on overlays
OVERLAYHTML=*variable-list*	specifies URLS to associate with primary chart overlay points
OVERLAYID=*variable-list*	specifies labels for primary chart overlay points
OVERLAYLEGLAB='*label*'	specifies label for overlay legend
OVERLAYSYM=*symbol-list*	specifies symbols for primary chart overlays
OVERLAYSYMHT=*value-list*	specifies symbol heights for primary chart overlays
WOVERLAY=*value-list*	specifies widths of primary chart overlay line segments
WOVERLAY2=*value-list*	specifies widths of secondary chart overlay line segments

Details

Constructing Charts for Means and Ranges

The following notation is used in this section:

μ	process mean (expected value of the population of measurements)
σ	process standard deviation (standard deviation of the population of measurements)
$\bar{X}_i$	mean of measurements in i^{th} subgroup
R_i	range of measurements in i^{th} subgroup
n_i	sample size of i^{th} subgroup
N	number of subgroups
$\overline{\overline{X}}$	weighted average of subgroup means
$d_2(n)$	expected value of the range of n independent normally distributed variables with unit standard deviation
$d_3(n)$	standard error of the range of n independent observations from a normal population with unit standard deviation
z_p	$100p^{\text{th}}$ percentile of the standard normal distribution
$D_p(n)$	$100p^{\text{th}}$ percentile of the distribution of the range of n independent observations from a normal population with unit standard deviation

Plotted Points

Each point on the $\bar{X}$ chart indicates the value of a subgroup mean ($\bar{X}_i$). For example, if the tenth subgroup contains the values 12, 15, 19, 16, and 14, the mean plotted for this subgroup is

$$\bar{X}_{10} = \frac{12 + 15 + 19 + 16 + 14}{5} = 15.2$$

Each point on the R chart indicates the value of a subgroup range (R_i). For example, the range plotted for the tenth subgroup is $R_{10} = 19 - 12 = 7$.

Central Lines

On an $\bar{X}$ chart, by default, the central line indicates an estimate of μ, which is computed as

$$\hat{\mu} = \overline{\overline{X}} = \frac{n_1\bar{X}_1 + \cdots + n_N\bar{X}_N}{n_1 + \cdots + n_N}$$

If you specify a known value (μ_0) for μ, the central line indicates the value of μ_0.

On an R chart, by default, the central line for the i^{th} subgroup indicates an estimate for the expected value of R_i, which is computed as $d_2(n_i)\hat{\sigma}$, where $\hat{\sigma}$ is an estimate of σ. If you specify a known value (σ_0) for σ, the central line indicates the value of $d_2(n_i)\sigma_0$. Note that the central line varies with n_i.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard errors of $\bar{X}_i$ and R_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that $\bar{X}_i$ or R_i exceeds the limits

The following table provides the formulas for the limits:

Table 50.23. Limits for $\bar{X}$ and R Charts

Control Limits	
$\bar{X}$ Chart	LCL = lower limit = $\overline{\overline{X}} - k\hat{\sigma}/\sqrt{n_i}$ UCL = upper limit = $\overline{\overline{X}} + k\hat{\sigma}/\sqrt{n_i}$
R Chart	LCL = lower limit = $\max(d_2(n_i)\hat{\sigma} - kd_3(n_i)\hat{\sigma}, 0)$ UCL = upper limit = $d_2(n_i)\hat{\sigma} + kd_3(n_i)\hat{\sigma}$

Probability Limits	
$\bar{X}$ Chart	LCL = lower limit = $\overline{\overline{X}} - z_{\alpha/2}(\hat{\sigma}/\sqrt{n_i})$ UCL = upper limit = $\overline{\overline{X}} + z_{\alpha/2}(\hat{\sigma}/\sqrt{n_i})$
R Chart	LCL = lower limit = $D_{\alpha/2}\hat{\sigma}$ UCL = upper limit = $D_{1-\alpha/2}\hat{\sigma}$

The formulas for R charts assume that the data are normally distributed. If standard values μ_0 and σ_0 are available for μ and σ, respectively, replace $\overline{\overline{X}}$ with μ_0 and $\hat{\sigma}$ with σ_0 in Table 50.23. Note that the limits vary with n_i and that the probability limits for R_i are asymmetric around the central line.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify μ_0 with the MU0= option or with the variable _MEAN_ in a LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables are saved:

Table 50.24. OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
_CP_	capability index C_p
CPK	capability index C_{pk}
CPL	capability index CPL
CPM	capability index C_{pm}
CPU	capability index CPU
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLR	lower control limit for subgroup range
LCLX	lower control limit for subgroup mean
LIMITN	nominal sample size associated with the control limits
LSL	lower specification limit
MEAN	process mean ($\overline{\overline{X}}$ or μ_0)
_R_	value of central line on R chart
SIGMAS	multiple (k) of standard error of $\bar{X}_i$ or R_i
STDDEV	process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*subgroup-variable* specified in the XRCHART statement
TARGET	target value
TYPE	type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLR	upper control limit for subgroup range
UCLX	upper control limit for subgroup mean
USL	upper specification limit
VAR	*process* specified in the XRCHART statement

Notes:

1. If the control limits vary with subgroup sample size, the special missing value *V* is assigned to the variables _LIMITN_, _LCLX_, _UCLX_, _LCLR_, _R_, and _UCLR_.
2. If the limits are defined in terms of a multiple k of the standard errors of $\bar{X}_i$ and R_i, the value of _ALPHA_ is computed as $\alpha = 2(1 - \Phi(k))$, where $\Phi(\cdot)$ is the standard normal distribution function.
3. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = \Phi^{-1}(1 - \alpha/2)$, where Φ^{-1} is the inverse standard normal distribution function.
4. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you

provide a target value with the TARGET= option. See "Capability Indices" on page 1774 for computational details.

5. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the XRCHART statement. For an example, see "Saving Control Limits" on page 1745.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables are saved:

- the *subgroup-variable*
- a subgroup mean variable named by *process* suffixed with *X*
- a subgroup range variable named by *process* suffixed with *R*
- a subgroup sample size variable named by *process* suffixed with *N*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Variables containing subgroup means, ranges, and sample sizes are created for each *process* specified in the XRCHART statement. For example, consider the following statements:

```
proc shewhart data=steel;
   xrchart (width diameter)*lot / outhistory=summary;
run;
```

The data set SUMMARY contains variables named LOT, WIDTHX, WIDTHR, WIDTHN, DIAMTERX, DIAMTERR, and DIAMTERN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 1744.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables are saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on $\bar{X}$ chart
EXLIMR	control limit exceeded on R chart
LCLR	lower control limit for range
LCLX	lower control limit for mean
LIMITN	nominal sample size associated with the control limits
MEAN	process mean
_R_	average range
SIGMAS	multiple (k) of the standard error associated with control limits
subgroup	values of the subgroup variable
SUBN	subgroup sample size
SUBR	subgroup range
SUBX	subgroup mean
TESTS	tests for special causes signaled on $\bar{X}$ chart
TESTS2	tests for special causes signaled on R chart
UCLR	upper control limit for range
UCLX	upper control limit for mean
VAR	*process* specified in the XRCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved, depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The k^{th} character of a value of _TESTS_ is k if Test k is positive at that subgroup. For example, if you request all eight tests and Tests 2 and 8 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, an 8 for the eighth character, and blanks for the other six characters.
3. The variable _TESTS2_ is saved if you specify the TESTS2= option.
4. The variables _EXLIM_, _EXLIMR_, _TESTS_, and _TESTS2_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1745.

ODS Tables

The following table summarizes the ODS tables that you can request with the XRCHART statement.

Table 50.25. ODS Tables Produced with the XRCHART Statement

Table Name	Description	Options
XRCHART	$\bar{X}$ and R chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the XRCHART statement must be a SAS variable in the DATA= data set. This variable provides measurements which must be grouped into subgroup samples indexed by the *subgroup-variable*. The *subgroup-variable*, which is specified in the XRCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. If the t^{th} subgroup contains n_i items, there should be n_i consecutive observations for which the value of the subgroup variable is the index of the t^{th} subgroup. For example, if each subgroup contains five items and there are 30 subgroup samples, the DATA= data set should contain 150 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all the observations in a DATA= data set. However, if the DATA= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 1936).

For an example of a DATA= data set, see "Creating Charts for Means and Ranges from Raw Data" on page 1738.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set CONLIMS:*

```
proc shewhart data=info limits=conlims;
   xrchart weight*batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLX_, _MEAN_, _UCLX_, _LCLR_, _R_, and _UCLR_, which specify the control limits directly
- the variables _MEAN_ and _STDDEV_, which are used to calculate the control limits according to the equations in Table 50.23 on page 1763

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are `ESTIMATE`, `STANDARD`, `STDMU`, and `STDSIGMA`.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1748.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART, CUSUM, or MACONTROL procedures or to read output data sets created with SAS summarization procedures, such as PROC MEANS.

*In Release 6.09 and in earlier releases, it is necessary to specify the READLIMITS option.

A HISTORY= data set used with the XRCHART statement must contain the following variables:

- the *subgroup-variable*
- a subgroup mean variable for each *process*
- a subgroup range variable for each *process*
- a subgroup sample size variable for each *process*

The names of the subgroup mean, subgroup range, and subgroup sample size variables must be the *process* name concatenated with the special suffix characters X, R, and N, respectively.

For example, consider the following statements:

```
proc shewhart history=summary;
   xrchart (weight yldstren)*batch;
run;
```

The data set SUMMARY must include the variables BATCH, WEIGHTX, WEIGHTR, WEIGHTN, YLDSRENX, YLDSRENR, and YLDSRENN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 1936 for an example).

For an example of a HISTORY= data set, see "Creating Charts for Means and Ranges from Summary Data" on page 1741.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure or to read data sets created by other SAS procedures. Because the SHEWHART procedure simply displays the information read from a TABLE= data set, you can use TABLE= data sets

to create specialized control charts. xamples are provided in Chapter 56, "Specialized Control Charts."

The following table lists the variables required in a TABLE= data set used with the XRCHART statement:

Table 50.26. Variables Required in a TABLE= Data Set

Variable	Description
LCLR	lower control limit for range
LCLX	lower control limit for mean
LIMITN	nominal sample size associated with the control limits
MEAN	process mean
_R_	average range
subgroup-variable	values of the *subgroup-variable*
SUBN	subgroup sample size
SUBR	subgroup range
SUBX	subgroup mean
UCLR	upper control limit for range
UCLX	upper control limit for mean

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes for subgroup means and must be a character variable of length 8.
- _TESTS2_ (if the TESTS2= option is specified). This variable is used to flag tests for special causes for subgroup ranges and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1745.

Subgroup Variables

The values of the *subgroup-variable*, which is specified in the chart statement, indicate how the observations in the input data set (a DATA=, HISTORY=, or TABLE= data set) are arranged into rational subgroups.* Typically, the values of the *subgroup-variable* are one of the following:

- *indices* that give the order in which subgroup samples were collected (for example, 1, 2, 3, . . .). An unformatted numeric *subgroup-variable* is appropriate for this situation. For an example using this type of *subgroup-variable*, see "Creating Charts for Means and Ranges from Raw Data" on page 1738.
- the *dates* or *times* at which subgroup samples were collected (for example, 01JUN, 02JUN, 03JUN, . . .). A numeric *subgroup-variable* with a SAS date, time, or datetime format is appropriate for this situation. You can optionally associate a format with the *subgroup-variable* by using a FORMAT statement; refer to *SAS Language Reference: Dictionary* for details. For an example using this type of *subgroup-variable*, see Example 50.3 on page 1781.
- *labels* that uniquely identify subgroup samples (for example, LOT39, LOTX12, LOT43A). A character *subgroup-variable* (with or without a format) is appropriate for this situation. For an example using this type of *subgroup-variable*, see Example 50.1 on page 1777.

The values of the *subgroup-variable* also determine how the horizontal axis of the control chart is scaled and labeled.

The notion of a rational subgroup is fundamental to the application of a Shewhart chart. You should select your subgroups so that if special causes of variation are present, the opportunity for variation within subgroups is minimized while the opportunity for variation between subgroups is maximized. In other words, the conditions within a subgroup should be homogeneous. The reason for this requirement is that the construction of the control limits is based on within-subgroup variability. Refer to Montgomery (1996) and Wheeler and Chambers (1986) for approaches to rational subgrouping.

The selection of subgroups is both a practical and a statistical issue that requires knowledge of the process and the sampling or measurement procedure. The values of the subgroup-variable should reflect the selection of subgroups and should not be assigned arbitrarily. Incorrect subgrouping or assignment of subgroup-variable values can result in control limits that are too tight or too wide.

If the input data set is a HISTORY= or TABLE= data set, each observation represents a distinct subgroup, and, consequently, the observations within each BY group must have distinct subgroup variable values. Similarly, if the input data set is a DATA= data set and you are using the CCHART, IRCHART, NPCHART, PCHART, or UCHART statement, each observation represents a distinct subgroup, and, consequently, the observations within each BY group must have distinct subgroup variable

*This discussion also applies to the use of *subgroup-variables* in the CUSUM procedure and the MACONTROL procedure.

values. However, if the input data set is a DATA= data set and you are using the BOXCHART, MCHART, MRCHART, RCHART, SCHART, XCHART, XRCHART, or XSCHART statement, subgroups are identified by groups of consecutive observations with identical values of the subgroup-variable.

The order of the observations in the input data set and the scaling of the horizontal axis depend on the type of the subgroup-variable, which can be numeric or character.

Numeric Subgroup Variables

If the subgroup-variable is numeric, the observations must be sorted in increasing order of the values of the subgroup variable. If you use a BY statement, first sort by the BY variables and then by the subgroup variable.

The unformatted values of the subgroup-variable are used to scale the horizontal axis of the control chart, and the formatted values are used to label the major tick marks on the horizontal axis. As a result, the horizontal distance between two points corresponding to consecutive subgroups is proportional to the difference between their unformatted subgroup values.

If a DATE, DATETIME, WEEKDATE, or WORDDATE format is associated with the subgroup variable, the major tick mark labels are split and displayed in two levels to save space. You can override this default with the TURNHLABELS option (which turns the labels vertically) or with tick label options in an AXISn statement specified with the HAXIS= option.

Character Subgroup Variables

If the subgroup-variable is numeric, the order of the observations is not checked. The horizontal axis is scaled so that the subgroups are spaced uniformly. Formatted subgroup variable values are used to label the major tick marks.

You can use a character subgroup variable to avoid gaps between groups of points or time values on a control chart. You can also use a character subgroup variable to create a chart in which the order of the points depends only on the order in which the subgroups are arranged in the input data set.

You should verify the order of the observations in the input data set before using a character subgroup variable in conjunction with the TESTS= option. With the exception of Test 1, the tests for special causes are applicable only if the subgroups are provided in chronological order. See Chapter 55, "Tests for Special Causes," for details.

To avoid collision of adjacent tick labels on the horizontal axis, the labels are thinned by default. You can override this default with the TURNHLABELS option or with tick label options in an AXISn statement specified with the HAXIS= option.

Methods for Estimating the Standard Deviation

When control limits are determined from the input data, three methods (referred to as default, MVLUE, and MVGRANGE) are available for estimating σ.

Default Method

The default estimate for σ is

$$\hat{\sigma} = \frac{R_1/d_2(n_1) + \cdots + R_N/d_2(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, and R_i is the sample range of the observations $x_{i1}, \ldots, x_{in_i}$ in the i^{th} subgroup.

$$R_i = \max_{1 \leq j \leq n_i} (x_{ij}) - \min_{1 \leq j \leq n_i} (x_{ij})$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$. The unbiasing factor $d_2(n_i)$ is defined so that, if the observations are normally distributed, the expected value of R_i is $d_2(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the *ASTM Manual on Presentation of Data and Control Chart Analysis* (1976).

MVLUE Method

If you specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). The MVLUE is a weighted average of N unbiased estimates of σ of the form $R_i/d_2(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{f_1 R_1/d_2(n_1) + \cdots + f_N R_N/d_2(n_N)}{f_1 + \cdots + f_N}$$

where

$$f_i = \frac{[d_2(n_i)]^2}{[d_3(n_i)]^2}$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The unbiasing factor $d_3(n_i)$ is defined so that, if the observations are normally distributed, the expected value of σ_{R_i} is $d_3(n_i)\sigma$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

MVGRANGE Method

If you specify SMETHOD=MVGRANGE, σ is estimated using a moving range of subgroup averages. This is appropriate for constructing control charts for means when the jth measurement in the ith subgroup can be modeled as $x_{ij} = \sigma_B \omega_i + \sigma_W \epsilon_{ij}$, where σ_B^2 is the between-subgroup variance, σ_W^2 is the within-subgroup variance, the ω_i are independent with zero mean and unit variance, and the ω_i are independent of the ϵ_{ij}.

The estimate for σ is

$$\hat{\sigma} = \bar{R}/d_2(n)$$

where $\bar{R}$ is the average of the moving ranges, n is the number of consecutive subgroup averages used to compute each moving range, and the unbiasing factor $d_2(n)$ is defined so that if the subgroup averages are normally distributed, the expected value of R_i is

$$E(R_i) = d_2(n_i)\sigma$$

This method is appropriate for constructing the three-way control chart that is advocated for this situation by Wheeler (1995). A three-way control chart is useful when sampling, or *within-group* variation is not the only source of variation, as discussed in "Multiple Components of Variation" on page 2009. A three-way control chart comprises a chart of subgroup means, a moving range chart of the subgroup means, and a chart of subgroup ranges. When you specify the SMETHOD=MVGRANGE option, the XRCHART statement produces the appropriate charts of subgroup means and subgroup ranges.

Capability Indices

This section provides formulas for process capability indices, which are saved in the OUTLIMITS= data set when you use the LSL= and USL= options to provide lower and upper specification limits (LSL and USL, respectively) for the *process*. The estimate $\hat{\sigma}$ is computed as described in the previous section, "Methods for Estimating the Standard Deviation."

The Index Cp

The process capability index C_p is computed as

$$C_p = (USL - LSL)/6\hat{\sigma}$$

If you do not specify both LSL and USL, the variable _CP_ is assigned a missing value.

The Index CPL

The process capability index *CPL* is computed as

$$CPL = (\overline{\overline{X}} - LSL)/3\hat{\sigma}$$

If you do not specify LSL, the variable _CPL_ is assigned a missing value.

The Index CPU

The process capability index *CPU* is computed as

$$CPU = (USL - \overline{\overline{X}})/3\hat{\sigma}$$

If you do not specify USL, the variable _CPU_ is assigned a missing value.

The Index Cpk

The process capability index C_{pk} is computed as

$$C_{pk} = \min(USL - \overline{\overline{X}}, \overline{\overline{X}} - LSL)/3\hat{\sigma}$$

If you specify only USL, the index C_{pk} is computed as

$$C_{pk} = (USL - \overline{\overline{X}})/3\hat{\sigma}$$

and if you specify only LSL, the index C_{pk} is computed as

$$C_{pk} = (\overline{\overline{X}} - LSL)/3\hat{\sigma}$$

The Index Cpm

The process capability index C_{pm} is computed as

$$C_{pm} = \frac{\min(T - LSL, USL - T)}{3\sqrt{\hat{\sigma}^2 + (\overline{\overline{X}} - T)^2}}$$

where T is the target value specified with the TARGET= option.

When a single specification limit (SL) and target are specified, C_{pm} is computed as

$$C_{pm} = \frac{|T - SL|}{3\sqrt{\hat{\sigma}^2 + (\overline{\overline{X}} - T)^2}}$$

You can also use the CAPABILITY procedure to compute a variety of capability indices. The SHEWHART procedure and the CAPABILITY procedure use the same formulas to calculate the indices, but they use different estimates for the process standard deviation σ.

- The SHEWHART procedure calculates $\hat{\sigma}$ from subgroup estimates of σ. For details, see the previous section, "Methods for Estimating the Standard Deviation."
- The CAPABILITY procedure calculates $\hat{\sigma}$ as the sample standard deviation of the entire sample. For details, see "Standard Deviation" on page 193.

Regardless of which method you use, you should verify that the process is in statistical control before interpreting the indices, and you should verify that the data are normally distributed. The CAPABILITY procedure provides a variety of statistical and graphical tests for checking normality.

Some references use different notation and names for capability indices. For example, the manual *Fundamental Statistical Process Control: Reference Manual* (1991) uses the term "process capability indices" for the indices listed in this section, and it uses the term "process performance indices" for the indices computed by the CAPABILITY procedure.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical ($\bar{X}$ chart)	DATA=	*process*
Vertical ($\bar{X}$ chart)	HISTORY=	subgroup mean variable
Vertical ($\bar{X}$ chart)	TABLE=	_SUBX_

You can specify distinct labels for the vertical axes of the $\bar{X}$ and R charts by breaking the vertical axis into two parts with a split character. Specify the split character with the SPLIT= option. The first part labels the vertical axis of the $\bar{X}$ chart, and the second part labels the vertical axis of the R chart.

For example, the following sets of statements specify the label *Avg Diameter in mm* for the vertical axis of the $\bar{X}$ chart and the label *Range in mm* for the vertical axis of the R chart:

```
proc shewhart data=wafers;
   xrchart diamtr*batch / split = '/' ;
   label diamtr = 'Avg Diameter in mm/Range in mm';
run;

proc shewhart history=wafersum;
   xrchart diamtr*batch / split = '/' ;
   label diamtrx = 'Avg Diameter in mm/Range in mm';
run;

proc shewhart table=wtable;
   xrchart diamtr*batch / split = '/' ;
   label _SUBX_ = 'Avg Diameter in mm/Range in mm';
run;
```

In this example, the label assignments are in effect only for the duration of the procedure step, and they temporarily override any permanent labels associated with the variables.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples

The SHEWHART Procedure

This section provides advanced examples of the XRCHART statement.

Example 50.1. Applying Tests for Special Causes

This example illustrates how you can apply tests for special causes to make $\bar{X}$ and R charts more sensitive to special causes of variation.

See SHWXR2 in the SAS/QC Sample Library

The weight of a roll of tape is measured before and after an adhesive is applied. The difference in weight represents the amount of adhesive applied to the tape during the coating process. The following data set contains the average and the range of the adhesive amounts for 21 samples of five rolls:

```
data tape;
   input sample $ weightx weightr;
   weightn=5;
   label weightx = 'Average Adhesive Amount'
         sample  = 'Sample Code';
   datalines;
C9  1270  35
C4  1258  25
A7  1248  24
A1  1260  39
A5  1273  29
D3  1260  21
D6  1259  37
D1  1240  37
R4  1260  28
H7  1255  19
H2  1268  36
H6  1253  36
P4  1273  29
P9  1275  22
J7  1257  24
J2  1269  41
J3  1249  36
B2  1264  31
G4  1258  25
G6  1248  36
G3  1248  30
;
run;
```

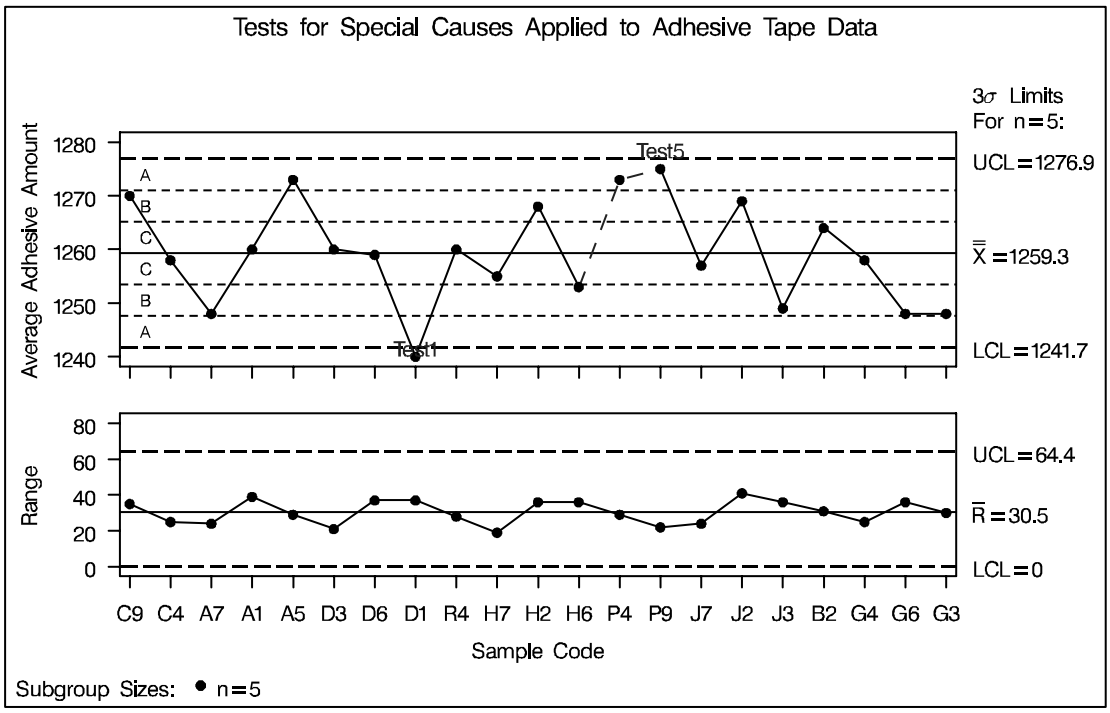
The following statements create $\bar{X}$ and R charts, apply several tests to the $\bar{X}$ chart, and tabulate the results:

```
title 'Tests for Special Causes Applied to Adhesive Tape Data';
proc shewhart history=tape;
   xrchart weight*sample / tests   = 1 to 5
                           tabletests
                           zonelabels
                           ltests  = 20;
run;
```

The charts are shown in Output 50.1.1, and the table is shown in Output 50.1.2. The TESTS= option requests Tests 1, 2, 3, 4, and 5, which are described in Chapter 55, "Tests for Special Causes." The TABLETESTS option requests a basic table of subgroup statistics and control limits with a column indicating which subgroups tested positive for special causes.

The ZONELABELS option displays zone lines and zone labels on the $\bar{X}$ chart. The zones are used to define the tests. The LTESTS= option specifies the line type used to connect the points in a pattern for a test that is signaled.

Output 50.1.1. Tests for Special Causes Displayed on $\bar{X}$ and R Charts



Output 50.1.1 and Output 50.1.2 indicate that Test 1 is positive at sample D1 and Test 5 is positive at sample P9. Test 1 detects one point beyond Zone A (outside the control limits), and Test 5 detects two out of three points in a row in Zone A or beyond.

Output 50.1.2. Tabular Form of $\bar{X}$ and R Charts

Tests for Special Causes Applied to Adhesive Tape Data

Means and Ranges Chart Summary for weight

sample	Subgroup Sample Size	---3 Sigma Limits with n=5 for Mean--- Lower Limit	Subgroup Mean	Upper Limit	Special Tests Signaled
C9	5	1241.7065	1270.0000	1276.8650	
C4	5	1241.7065	1258.0000	1276.8650	
A7	5	1241.7065	1248.0000	1276.8650	
A1	5	1241.7065	1260.0000	1276.8650	
A5	5	1241.7065	1273.0000	1276.8650	
D3	5	1241.7065	1260.0000	1276.8650	
D6	5	1241.7065	1259.0000	1276.8650	
D1	5	1241.7065	1240.0000	1276.8650	1
R4	5	1241.7065	1260.0000	1276.8650	
H7	5	1241.7065	1255.0000	1276.8650	
H2	5	1241.7065	1268.0000	1276.8650	
H6	5	1241.7065	1253.0000	1276.8650	
P4	5	1241.7065	1273.0000	1276.8650	
P9	5	1241.7065	1275.0000	1276.8650	5
J7	5	1241.7065	1257.0000	1276.8650	
J2	5	1241.7065	1269.0000	1276.8650	
J3	5	1241.7065	1249.0000	1276.8650	
B2	5	1241.7065	1264.0000	1276.8650	
G4	5	1241.7065	1258.0000	1276.8650	
G6	5	1241.7065	1248.0000	1276.8650	
G3	5	1241.7065	1248.0000	1276.8650	

Means and Ranges Chart Summary for weight

sample	-3 Sigma Limits with n=5 for Range- Lower Limit	Subgroup Range	Upper Limit
C9	0	35.000000	64.441879
C4	0	25.000000	64.441879
A7	0	24.000000	64.441879
A1	0	39.000000	64.441879
A5	0	29.000000	64.441879
D3	0	21.000000	64.441879
D6	0	37.000000	64.441879
D1	0	37.000000	64.441879
R4	0	28.000000	64.441879
H7	0	19.000000	64.441879
H2	0	36.000000	64.441879
H6	0	36.000000	64.441879
P4	0	29.000000	64.441879
P9	0	22.000000	64.441879
J7	0	24.000000	64.441879
J2	0	41.000000	64.441879
J3	0	36.000000	64.441879
B2	0	31.000000	64.441879
G4	0	25.000000	64.441879
G6	0	36.000000	64.441879
G3	0	30.000000	64.441879

Example 50.2. Specifying Standard Values for the Process Mean and Standard Deviation

See SHWXR3 in the SAS/QC Sample Library

By default, the XRCHART statement estimates the process mean (μ) and standard deviation (σ) from the data, as in the previous example. However, there are applications in which standard values (μ_0 and σ_0) are available based, for instance, on previous experience or extensive sampling. You can specify these values with the MU0= and SIGMA0= options.

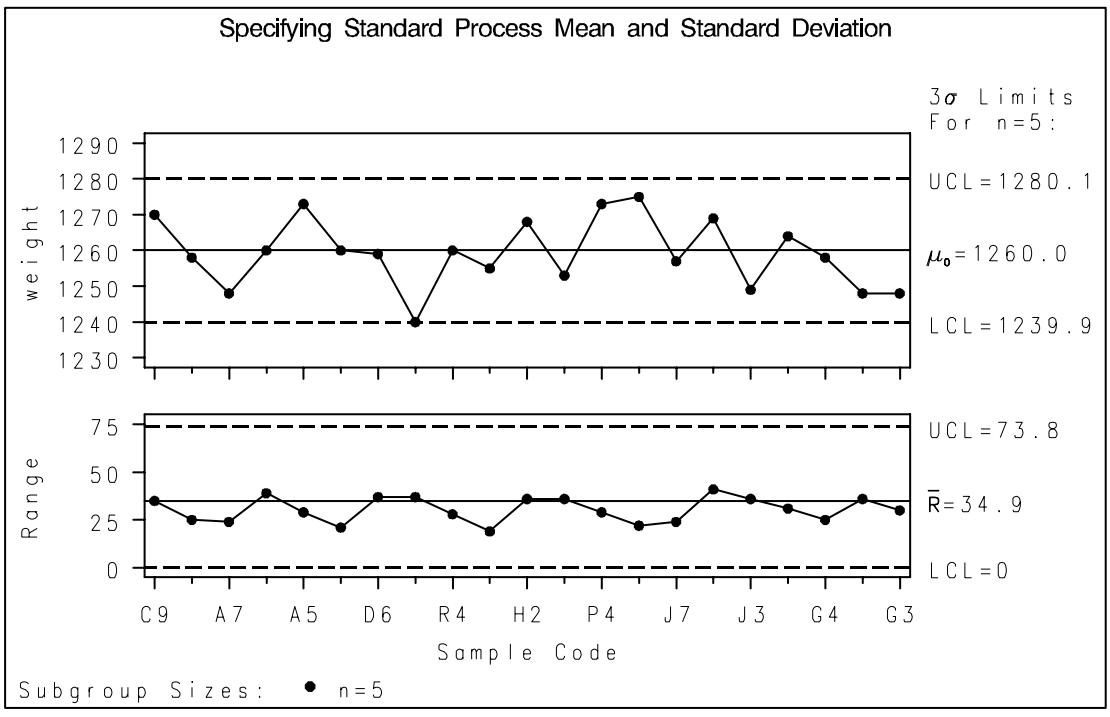
For example, suppose it is known that the adhesive coating process introduced in the previous example has a mean of 1260 and standard deviation of 15. The following statements specify these standard values:

```
title 'Specifying Standard Process Mean and Standard Deviation';
proc shewhart history=tape;
   xrchart weight*sample / mu0     = 1260
                           sigma0  = 15
                           font    = 'Lucida Console'
                           xsymbol = mu0;
run;
```

Here the FONT= option is used to specify the name of a hardware font to be used for these charts. In this case the requested font is Lucida Console, a Windows TrueType font. See *SAS/GRAPH Software: Reference* and *SAS Companion for Microsoft Windows* for more information on hardware and TrueType fonts.

The XSYMBOL= option specifies the label for the central line on the $\bar{X}$ chart. The resulting $\bar{X}$ and R charts are shown in Output 50.2.1.

Output 50.2.1. Specifying Standard Values with MU0= and SIGMA0=



The central lines and control limits for both charts are determined using μ_0 and σ_0 (see the equations in Table 50.23 on page 1763). Output 50.2.1 indicates that the process is in statistical control.

You can also specify μ_0 and σ_0 with the variables _MEAN_ and _STDDEV_ in a LIMITS= data set, as illustrated by the following statements:

```
data tapelim;
   length _var_ _subgrp_ _type_ $8;
   _var_    = 'weight';
   _subgrp_ = 'sample';
   _type_   = 'STANDARD';
   _limitn_ = 5;
   _mean_   = 1260;
   _stddev_ = 15;

proc shewhart history=tape limits=tapelim;
   xrchart weight*sample / xsymbol=mu0;
run;
```

The variables _VAR_ and _SUBGRP_ are required, and their values must match the *process* and *subgroup-variable*, respectively, specified in the XRCHART statement. The bookkeeping variable _TYPE_ is not required, but it is recommended to indicate that the variables _MEAN_ and _STDDEV_ provide standard values rather than estimated values.

The resulting charts (not shown here) are identical to those shown in Output 50.2.1.

Example 50.3. Working with Unequal Subgroup Sample Sizes

See SHWXR4 in the SAS/QC Sample Library

The following data set (WIRE) contains breaking strength measurements recorded in pounds per inch for 25 samples from a metal wire manufacturing process. The subgroup sample sizes vary between 3 and 7.

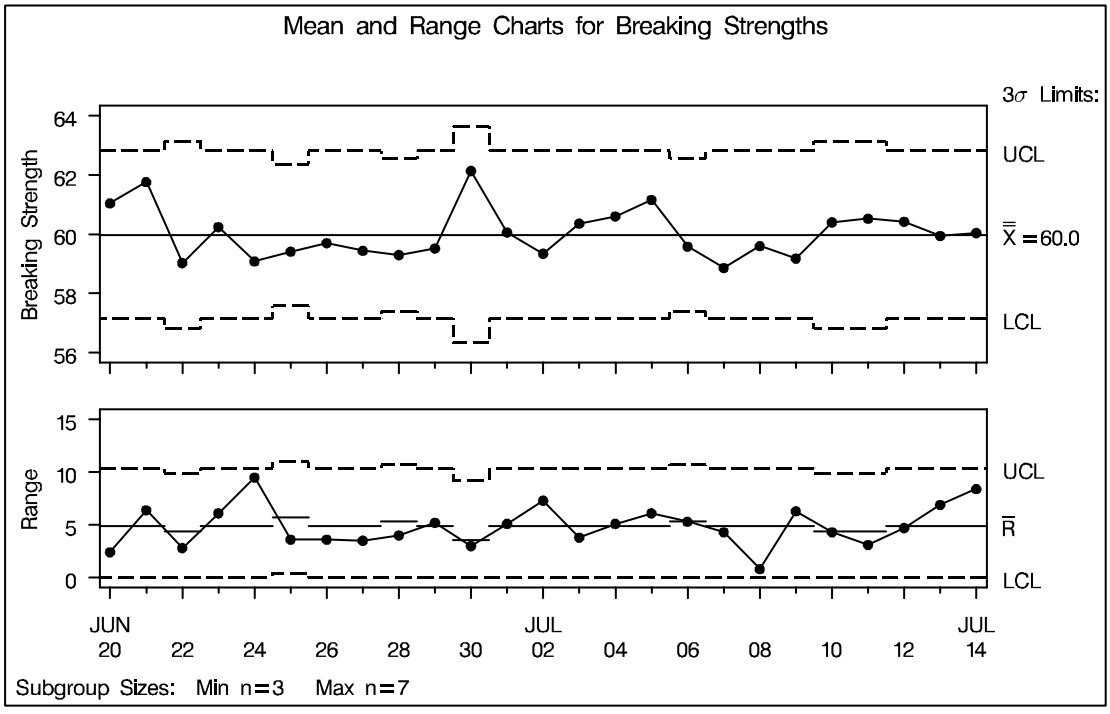
```
data wire;
   input day size @;
   informat day date7.;
   format day date7.;
   do i=1 to size;
      input brstr @@;
      output;
      end;
   drop i size;
   label brstr  ='Breaking Strength';
   datalines;
20JUN94 5 60.6 62.3 62.0 60.4 59.9
21JUN94 5 61.9 62.1 60.6 58.9 65.3
22JUN94 4 57.8 60.5 60.1 57.7
23JUN94 5 56.8 62.5 60.1 62.9 58.9
24JUN94 5 63.0 60.7 57.2 61.0 53.5
25JUN94 7 58.7 60.1 59.7 60.1 59.1 57.3 60.9
26JUN94 5 59.3 61.7 59.1 58.1 60.3
27JUN94 5 61.3 58.5 57.8 61.0 58.6
28JUN94 6 59.5 58.3 57.5 59.4 61.5 59.6
```

```
29JUN94 5 61.7 60.7 57.2 56.5 61.5
30JUN94 3 63.9 61.6 60.9
01JUL94 5 58.7 61.4 62.4 57.3 60.5
02JUL94 5 56.8 58.5 55.7 63.0 62.7
03JUL94 5 62.1 60.6 62.1 58.7 58.3
04JUL94 5 59.1 60.4 60.4 59.0 64.1
05JUL94 5 59.9 58.8 59.2 63.0 64.9
06JUL94 6 58.8 62.4 59.4 57.1 61.2 58.6
07JUL94 5 60.3 58.7 60.5 58.6 56.2
08JUL94 5 59.2 59.8 59.7 59.3 60.0
09JUL94 5 62.3 56.0 57.0 61.8 58.8
10JUL94 4 60.5 62.0 61.4 57.7
11JUL94 4 59.3 62.4 60.4 60.0
12JUL94 5 62.4 61.3 60.5 57.7 60.2
13JUL94 5 61.2 55.5 60.2 60.4 62.4
14JUL94 5 59.0 66.1 57.7 58.5 58.9
;
run;
```

The following statements request $\bar{X}$ and R charts, shown in Output 50.3.1, for the strength measurements:

```
title 'Mean and Range Charts for Breaking Strengths';
proc shewhart data=wire;
   xrchart brstr*day / nohlabel ;
run;
```

Output 50.3.1. $\bar{X}$ and R Charts with Varying Subgroup Sample Sizes



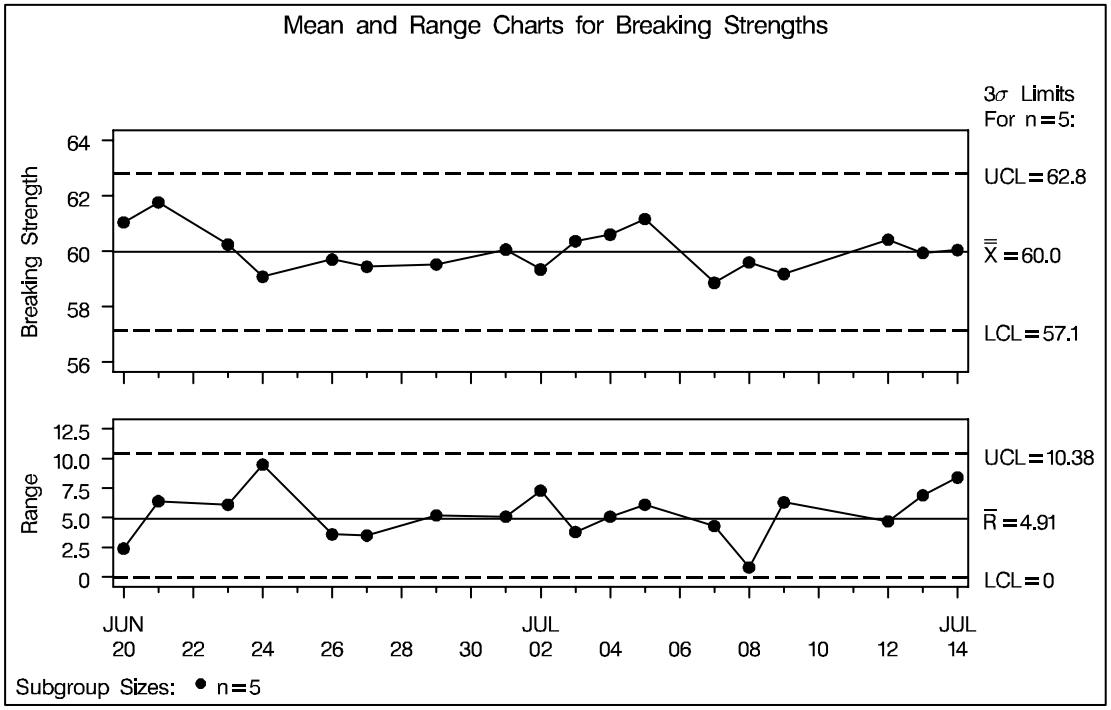
Note that the central line on the R chart and the control limits on both charts vary with the subgroup sample size. The sample size legend in the lower left corner displays the minimum and maximum subgroup sample sizes.

The XRCHART statement provides various options for working with unequal subgroup sample sizes. For example, you can use the LIMITN= option to specify a fixed (nominal) sample size for computing control limits, as illustrated by the following statements:

```
title 'Mean and Range Charts for Breaking Strengths';
proc shewhart data=wire;
   xrchart brstr*day / nohlabel
                       limitn   = 5;
run;
```

The resulting charts are shown in Output 50.3.2.

Output 50.3.2. Control Limits Based on Fixed Sample Size



Note that the only points displayed on the chart are those corresponding to subgroups whose sample sizes match the nominal sample size of five. To plot points for all subgroups (regardless of subgroup sample size), you can specify the ALLN option, as follows:

```
title 'Mean and Range Charts for Breaking Strengths';
proc shewhart data=wire;
   xrchart brstr*day / nohlabel
                       limitn = 5
                       alln
                       nmarkers;
run;
```

The charts are shown in Output 50.3.3. The NMARKERS option requests special symbols to identify points for which the subgroup sample size differs from the nominal sample size.

Output 50.3.3. Displaying All Subgroups Regardless of Sample Size

You can use the SMETHOD= option to determine how the process standard deviation σ is to be estimated when the subgroup sample sizes vary. The default method computes $\hat{\sigma}$ as an unweighted average of subgroup estimates of σ. Specifying SMETHOD=MVLUE requests an estimate that assigns greater weight to estimates of σ from subgroups with larger sample sizes. For more information, see "Methods for Estimating the Standard Deviation" on page 1773.

The following statements apply both methods:

```
proc shewhart data=wire;
   xrchart brstr*day / outlimits = wlim1
                       outindex  = 'Default'
                       nochart;
   xrchart brstr*day / smethod   = mvlue
                       outlimits = wlim2
                       outindex  = 'MVLUE'
                       nochart;
run;

data wlimits;
   set wlim1 wlim2;
run;
```

The data set WLIMITS is listed in Output 50.3.4.

Output 50.3.4. Listing of the Data Set WLIMITS

The WLIMITS Data Set

VAR	_SUBGRP_	_INDEX_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_	_MEAN_	_UCLX_	_LCLR_	_R_	_UCLR_	_STDDEV_
brstr	day	Default	ESTIMATE	V	.002699796	3	V	59.9766	V	V	V	V	2.11146
brstr	day	MVLUE	ESTIMATE	V	.002699796	3	V	59.9766	V	V	V	V	2.11240

The variables in an OUTLIMITS= data set whose values vary with subgroup sample size are assigned the special missing value *V*. Consequently, the control limit variables (_LCLX_, _UCLX_, _LCLR_, and _UCLR_), as well as the variables _R_ and _LIMITN_, have this value.

Chapter 51
XSCHART Statement

Chapter Contents

OVERVIEW 1789

GETTING STARTED 1790
- Creating Charts for Means and Standard Deviations from Raw Data 1790
- Creating Charts for Means and Standard Deviations from Summary Data 1793
- Saving Summary Statistics 1796
- Saving Control Limits 1797
- Reading Preestablished Control Limits 1800

SYNTAX 1801
- Summary of Options 1802

DETAILS 1814
- Constructing Charts for Means and Standard Deviations 1814
- Output Data Sets 1816
- ODS Tables 1819
- Input Data Sets 1819
- Methods for Estimating the Standard Deviation 1823
- Axis Labels 1825
- Missing Values 1825

EXAMPLES 1826
- Example 51.1. Specifying Probability Limits 1826
- Example 51.2. Computing Subgroup Summary Statistics 1827
- Example 51.3. Analyzing Nonnormal Process Data 1828

Chapter 51
XSCHART Statement

Overview

The XSCHART statement creates $\bar{X}$ and s charts for subgroup means and standard deviations, which are used to analyze the central tendency and variability of a process.

You can use options in the XSCHART statement to

- compute control limits from the data based on a multiple of the standard error of the plotted means and standard deviations or as probability limits
- tabulate subgroup sample sizes, subgroup means, subgroup standard deviations, control limits, and other information
- save control limits in an output data set
- save subgroup sample sizes, subgroup means, and subgroup standard deviations in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify a method for estimating the process standard deviation
- specify a known (standard) process mean and standard deviation for computing control limits
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the charts more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

Getting Started

This section introduces the XSCHART statement with simple examples that illustrate commonly used options. Complete syntax for the XSCHART statement is presented in the "Syntax" section on page 1801, and advanced examples are given in the "Examples" section on page 1826.

Creating Charts for Means and Standard Deviations from Raw Data

See SHWXS1 in the SAS/QC Sample Library

A petroleum company uses a turbine to heat water into steam, which is then pumped into the ground to make oil less viscous and easier to extract. This process occurs 20 times daily, and the amount of power (in kilowatts) used to heat the water to the desired temperature is recorded. The following statements create a SAS data set named TURBINE, which contains the power output measurements for 20 days:

```
data turbine;
   informat day date7.;
   format day date5.;
   input day @;
   do i=1 to 10;
      input kwatts @;
      output;
      end;
   drop i;
   datalines;
04JUL94 3196 3507 4050 3215 3583 3617 3789 3180 3505 3454
04JUL94 3417 3199 3613 3384 3475 3316 3556 3607 3364 3721
05JUL94 3390 3562 3413 3193 3635 3179 3348 3199 3413 3562
05JUL94 3428 3320 3745 3426 3849 3256 3841 3575 3752 3347
06JUL94 3478 3465 3445 3383 3684 3304 3398 3578 3348 3369
06JUL94 3670 3614 3307 3595 3448 3304 3385 3499 3781 3711
07JUL94 3448 3045 3446 3620 3466 3533 3590 3070 3499 3457
07JUL94 3411 3350 3417 3629 3400 3381 3309 3608 3438 3567
08JUL94 3568 2968 3514 3465 3175 3358 3460 3851 3845 2983
08JUL94 3410 3274 3590 3527 3509 3284 3457 3729 3916 3633
09JUL94 3153 3408 3741 3203 3047 3580 3571 3579 3602 3335
09JUL94 3494 3662 3586 3628 3881 3443 3456 3593 3827 3573
10JUL94 3594 3711 3369 3341 3611 3496 3554 3400 3295 3002
10JUL94 3495 3368 3726 3738 3250 3632 3415 3591 3787 3478
11JUL94 3482 3546 3196 3379 3559 3235 3549 3445 3413 3859
11JUL94 3330 3465 3994 3362 3309 3781 3211 3550 3637 3626
12JUL94 3152 3269 3431 3438 3575 3476 3115 3146 3731 3171
12JUL94 3206 3140 3562 3592 3722 3421 3471 3621 3361 3370
13JUL94 3421 3381 4040 3467 3475 3285 3619 3325 3317 3472
13JUL94 3296 3501 3366 3492 3367 3619 3550 3263 3355 3510
14JUL94 3795 3872 3559 3432 3322 3587 3336 3732 3451 3215
14JUL94 3594 3410 3335 3216 3336 3638 3419 3515 3399 3709
15JUL94 3850 3431 3460 3623 3516 3810 3671 3602 3480 3388
15JUL94 3365 3845 3520 3708 3202 3365 3731 3840 3182 3677
16JUL94 3711 3648 3212 3664 3281 3371 3416 3636 3701 3385
```

```
16JUL94 3769 3586 3540 3703 3320 3323 3480 3750 3490 3395
17JUL94 3596 3436 3757 3288 3417 3331 3475 3600 3690 3534
17JUL94 3306 3077 3357 3528 3530 3327 3113 3812 3711 3599
18JUL94 3428 3760 3641 3393 3182 3381 3425 3467 3451 3189
18JUL94 3588 3484 3759 3292 3063 3442 3712 3061 3815 3339
19JUL94 3746 3426 3320 3819 3584 3877 3779 3506 3787 3676
19JUL94 3727 3366 3288 3684 3500 3501 3427 3508 3392 3814
20JUL94 3676 3475 3595 3122 3429 3474 3125 3307 3467 3832
20JUL94 3383 3114 3431 3693 3363 3486 3928 3753 3552 3524
21JUL94 3349 3422 3674 3501 3639 3682 3354 3595 3407 3400
21JUL94 3401 3359 3167 3524 3561 3801 3496 3476 3480 3570
22JUL94 3618 3324 3475 3621 3376 3540 3585 3320 3256 3443
22JUL94 3415 3445 3561 3494 3140 3090 3561 3800 3056 3536
23JUL94 3421 3787 3454 3699 3307 3917 3292 3310 3283 3536
23JUL94 3756 3145 3571 3331 3725 3605 3547 3421 3257 3574
;
run;
```

A partial listing of TURBINE is shown in Figure 51.1.

Kilowatt Power Output Data

Obs	day	kwatts
1	04JUL	3196
2	04JUL	3507
3	04JUL	4050
.	.	.
.	.	.
.	.	.
21	05JUL	3390
22	05JUL	3562
23	05JUL	3413
.	.	.
.	.	.
.	.	.
398	23JUL	3421
399	23JUL	3257
400	23JUL	3574

Figure 51.1. Partial Listing of the Data Set TURBINE

The data set is said to be in "strung-out" form since each observation contains the day and power output for a single heating. The first 20 observations contain the power outputs for the first day, the second 20 observations contain the power outputs for the second day, and so on. Because the variable DAY classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable KWATTS contains the power output measurements and is referred to as the *process variable* (or *process* for short).

You can use $\bar{X}$ and s charts to determine whether the heating process is in control. The following statements create the $\bar{X}$ and s charts shown in Figure 51.2:

```
title 'Mean and Standard Deviation Charts for Power Output';
proc shewhart data=turbine;
   xschart kwatts*day ;
run;
```

This example illustrates the basic form of the XSCHART statement. After the keyword XSCHART, you specify the *process* to analyze (in this case KWATTS), followed by an asterisk and the *subgroup-variable* (DAY).

The input data set is specified with the DATA= option in the PROC SHEWHART statement.

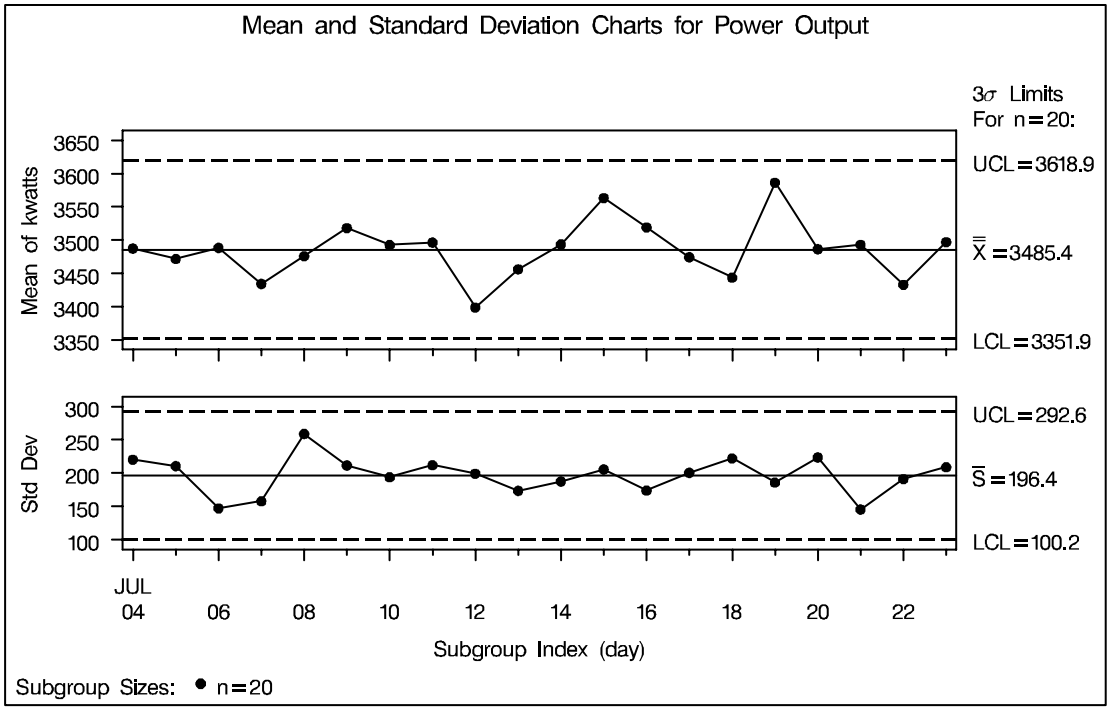


Figure 51.2. $\bar{X}$ and s Charts for Power Output Data

Each point on the $\bar{X}$ chart represents the mean of the measurements for a particular day. For instance, the mean plotted for the first day is $(3196+3507+\cdots+3721)/20 = 3487.4$.

Each point on the s chart represents the standard deviation of the measurements for a particular day. For instance, the standard deviation plotted for the first day is

$$\sqrt{\frac{(3196-3487.4)^2+(3507-3487.4)^2+\cdots+(3721-3487.4)^2}{19}} = 220.26$$

Since all the points lie within the control limits, it can be concluded that the process is in statistical control.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 51.23 on page 1815. You can also read control

limits from an input data set; see "Reading Preestablished Control Limits" on page 1800.

For computational details, see "Constructing Charts for Means and Standard Deviations" on page 1814. For more details on reading raw data, see "DATA= Data Set" on page 1819.

Creating Charts for Means and Standard Deviations from Summary Data

See SHWXS1 in the SAS/QC Sample Library

The previous example illustrates how you can create $\bar{X}$ and s charts using raw data (process measurements). However, in many applications the data are provided as subgroup summary statistics. This example illustrates how you can use the XSCHART statement with data of this type.

The following data set (OILSUM) provides the data from the preceding example in summarized form:

```
data oilsum;
   input day kwattsx kwattss kwattsn;
   informat day date7. ;
   format day date5. ;
   label day='Date of Measurement';
   datalines;
04JUL94 3487.40 220.260 20
05JUL94 3471.65 210.427 20
06JUL94 3488.30 147.025 20
07JUL94 3434.20 157.637 20
08JUL94 3475.80 258.949 20
09JUL94 3518.10 211.566 20
10JUL94 3492.65 193.779 20
11JUL94 3496.40 212.024 20
12JUL94 3398.50 199.201 20
13JUL94 3456.05 173.455 20
14JUL94 3493.60 187.465 20
15JUL94 3563.30 205.472 20
16JUL94 3519.05 173.676 20
17JUL94 3474.20 200.576 20
18JUL94 3443.60 222.084 20
19JUL94 3586.35 185.724 20
20JUL94 3486.45 223.474 20
21JUL94 3492.90 145.267 20
22JUL94 3432.80 190.994 20
23JUL94 3496.90 208.858 20
;
run;
```

A partial listing of OILSUM is shown in Figure 51.3.

Summary Data Set for Power Output

day	kwattsx	kwattss	kwattsn
04JUL	3487.40	220.260	20
05JUL	3471.65	210.427	20
06JUL	3488.30	147.025	20
07JUL	3434.20	157.637	20
08JUL	3475.80	258.949	20
.	.	.	.
.	.	.	.
.	.	.	.

Figure 51.3. The Summary Data Set OILSUM

There is exactly one observation for each subgroup (note that the subgroups are still indexed by DAY). The variable KWATTSX contains the subgroup means, the variable KWATTSS contains the subgroup standard deviations, and the variable KWATTSN contains the subgroup sample sizes (which are all 20). You can read this data set by specifying it as a HISTORY= data set in the PROC SHEWHART statement, as follows:

```
title 'Mean and Standard Deviation Charts for Power Output';
proc shewhart history=oilsum lineprinter;
   xschart kwatts*day='*';
run;
```

The resulting $\bar{X}$ and s charts are shown in Figure 51.4. Since the LINEPRINTER option is specified in the PROC SHEWHART statement, line printer output is produced. The asterisk (*) specified in single quotes after the *subgroup-variable* indicates the character used to plot the points. This character must follow an equal sign.

Note that KWATTS is *not* the name of a SAS variable in the data set OILSUM but is, instead, the common prefix for the names of the three SAS variables KWATTSX, KWATTSS, and KWATTSN. The suffix characters *X*, *S*, and *N* indicate *mean*, *standard deviation*, and *sample size*, respectively. Thus, you can specify three subgroup summary variables in the HISTORY= data set with a single name (KWATTS), which is referred to as the *process*. The name DAY specified after the asterisk is the name of the *subgroup-variable*.

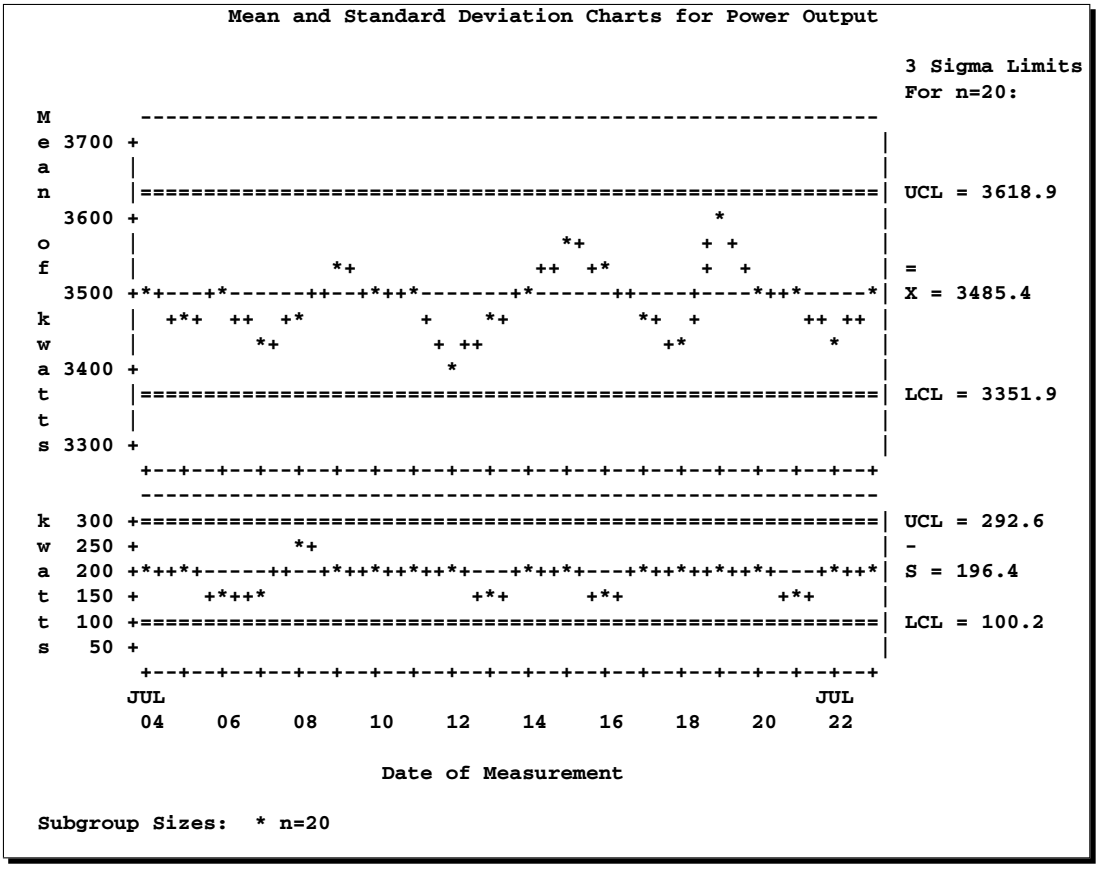


Figure 51.4. $\bar{X}$ and s Charts for Power Output Data

In general, a HISTORY= input data set used with the XSCHART statement must contain the following variables:

- subgroup variable
- subgroup mean variable
- subgroup standard deviation variable
- subgroup sample size variable

Furthermore, the names of the subgroup mean, standard deviation, and sample size variables must begin with the *process* name specified in the XSCHART statement and end with the special suffix characters *X*, *S*, and *N*, respectively. If the names do not follow this convention, you can use the RENAME option to rename the variables for the duration of the SHEWHART procedure step. For an illustration, see Example 51.2 on page 1827.

In summary, the interpretation of *process* depends on the input data set:

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.

- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names of the variables containing the summary statistics.

For more information, see "HISTORY= Data Set" on page 1820.

Saving Summary Statistics

See SHWXS1 in the SAS/QC Sample Library

In this example, the XSCHART statement is used to create a summary data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read measurements from the data set TURBINE (see page 1790) and create a summary data set named TURBHIST:

```
proc shewhart data=turbine;
   xschart kwatts*day / outhistory = turbhist
                        nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the charts, which would be identical to those in Figure 51.2. Options such as OUTHISTORY= and NOCHART are specified after the slash (/) in the XSCHART statement. A complete list of options is presented in the "Syntax" section on page 1801.

Figure 51.5 contains a partial listing of TURBHIST.

Summary Data Set for Power Output

day	kwattsX	kwattsS	kwatts N
04JUL	3487.40	220.260	20
05JUL	3471.65	210.427	20
06JUL	3488.30	147.025	20
07JUL	3434.20	157.637	20
08JUL	3475.80	258.949	20
.	.	.	.
.	.	.	.
.	.	.	.

Figure 51.5. The Summary Data Set TURBHIST

There are four variables in the data set TURBHIST.

- DAY contains the subgroup index.
- KWATTSX contains the subgroup means.
- KWATTSS contains the subgroup standard deviations.
- KWATTSN contains the subgroup sample sizes.

Note that the summary statistic variables are named by adding the suffix characters *X*, *S*, and *N* to the *process* KWATTS specified in the XSCHART statement. In other

words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 1817.

Saving Control Limits

See SHWXS1 in the SAS/QC Sample Library

You can save the control limits for $\bar{X}$ and s charts in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1800) or modify the limits with a DATA step program.

The following statements read measurements from the data set TURBINE (see page 1790) and save the control limits displayed in Figure 51.2 in a data set named TURBLIM:

```
proc shewhart data=turbine;
   xschart kwatts*day / outlimits=turblim
                        nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the charts. The data set TURBLIM is listed in Figure 51.6.

Control Limits for Power Output Data

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_
kwatts	day	ESTIMATE	20	.002699796	3	3351.92

MEAN	_UCLX_	_LCLS_	_S_	_UCLS_	_STDDEV_
3485.41	3618.90	100.207	196.396	292.584	198.996

Figure 51.6. The Data Set TURBLIM Containing Control Limit Information

The data set TURBLIM contains one observation with the limits for *process* KWATTS. The variables _LCLX_ and _UCLX_ contain the lower and upper control limits for the $\bar{X}$ chart, and the variables _LCLS_ and _UCLS_ contain the lower and upper control limits for the s chart. The variable _MEAN_ contains the central line for the $\bar{X}$ chart, and the variable _S_ contains the central line for the s chart. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values. For more information, see "OUTLIMITS= Data Set" on page 1816.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=turbine;
   xschart kwatts*day / outtable=turbtab
                        nochart;
run;
```

The data set TURBTAB contains one observation for each subgroup sample. The variables _SUBX_, _SUBS_, and _SUBN_ contain the subgroup means, subgroup standard deviations, and subgroup sample sizes. The variables _LCLX_ and _UCLX_ contain the lower and upper control limits for the $\bar{X}$ chart. The variables _LCLS_ and _UCLS_ contain the lower and upper control limits for the s chart. The variable _MEAN_ contains the central line for the $\bar{X}$ chart. The variable _S_ contains the central line for the s chart. The variables _VAR_ and BATCH contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1818.

The data set TURBTAB is listed in Figure 51.7.

A data set created with the OUTTABLE= option can be read later as a TABLE= data set. For example, the following statements read TURBTAB and display charts (not shown here) identical to those in Figure 51.2:

```
title 'Mean and Standard Deviation Charts for Power Output';
proc shewhart table=turbtab;
   xschart kwatts*day;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see Chapter 56, "Specialized Control Charts,"). For more information, see "TABLE= Data Set" on page 1821.

Summary Statistics and Control Limit Information

VAR	day	_SIGMAS_	_LIMITN_	_SUBN_	_LCLX_	_SUBX_	_MEAN_
kwatts	04JUL	3	20	20	3351.92	3487.40	3485.41
kwatts	05JUL	3	20	20	3351.92	3471.65	3485.41
kwatts	06JUL	3	20	20	3351.92	3488.30	3485.41
kwatts	07JUL	3	20	20	3351.92	3434.20	3485.41
kwatts	08JUL	3	20	20	3351.92	3475.80	3485.41
kwatts	09JUL	3	20	20	3351.92	3518.10	3485.41
kwatts	10JUL	3	20	20	3351.92	3492.65	3485.41
kwatts	11JUL	3	20	20	3351.92	3496.40	3485.41
kwatts	12JUL	3	20	20	3351.92	3398.50	3485.41
kwatts	13JUL	3	20	20	3351.92	3456.05	3485.41
kwatts	14JUL	3	20	20	3351.92	3493.60	3485.41
kwatts	15JUL	3	20	20	3351.92	3563.30	3485.41
kwatts	16JUL	3	20	20	3351.92	3519.05	3485.41
kwatts	17JUL	3	20	20	3351.92	3474.20	3485.41
kwatts	18JUL	3	20	20	3351.92	3443.60	3485.41
kwatts	19JUL	3	20	20	3351.92	3586.35	3485.41
kwatts	20JUL	3	20	20	3351.92	3486.45	3485.41
kwatts	21JUL	3	20	20	3351.92	3492.90	3485.41
kwatts	22JUL	3	20	20	3351.92	3432.80	3485.41
kwatts	23JUL	3	20	20	3351.92	3496.90	3485.41

UCLX	_EXLIM_	_LCLS_	_SUBS_	_S_	_UCLS_	_EXLIMS_
3618.90		100.207	220.260	196.396	292.584	
3618.90		100.207	210.427	196.396	292.584	
3618.90		100.207	147.025	196.396	292.584	
3618.90		100.207	157.637	196.396	292.584	
3618.90		100.207	258.949	196.396	292.584	
3618.90		100.207	211.566	196.396	292.584	
3618.90		100.207	193.779	196.396	292.584	
3618.90		100.207	212.024	196.396	292.584	
3618.90		100.207	199.201	196.396	292.584	
3618.90		100.207	173.455	196.396	292.584	
3618.90		100.207	187.465	196.396	292.584	
3618.90		100.207	205.472	196.396	292.584	
3618.90		100.207	173.676	196.396	292.584	
3618.90		100.207	200.576	196.396	292.584	
3618.90		100.207	222.084	196.396	292.584	
3618.90		100.207	185.724	196.396	292.584	
3618.90		100.207	223.474	196.396	292.584	
3618.90		100.207	145.267	196.396	292.584	
3618.90		100.207	190.994	196.396	292.584	
3618.90		100.207	208.858	196.396	292.584	

Figure 51.7. The OUTTABLE= Data Set TURBTAB

Reading Preestablished Control Limits

See SHWXS1 in the SAS/QC Sample Library

In the previous example, the OUTLIMITS= data set TURBLIM saved control limits computed from the measurements in TURBINE. This example shows how these limits can be applied to new data. The following statements create $\bar{X}$ and s charts for new measurements in a data set named TURBINE2 (not listed here) using the control limits in TURBLIM:

```
title 'Mean and Standard Deviation Charts for Power Output';
proc shewhart data=turbine2 limits=turblim;
   xschart kwatts*day ;
run;
```

The charts are shown in Figure 51.8. The LIMITS= option in the PROC SHEWHART statement specifies the data set containing preestablished control limit information. By default,* this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name KWATTS
- the value of _SUBGRP_ matches the *subgroup-variable* name DAY

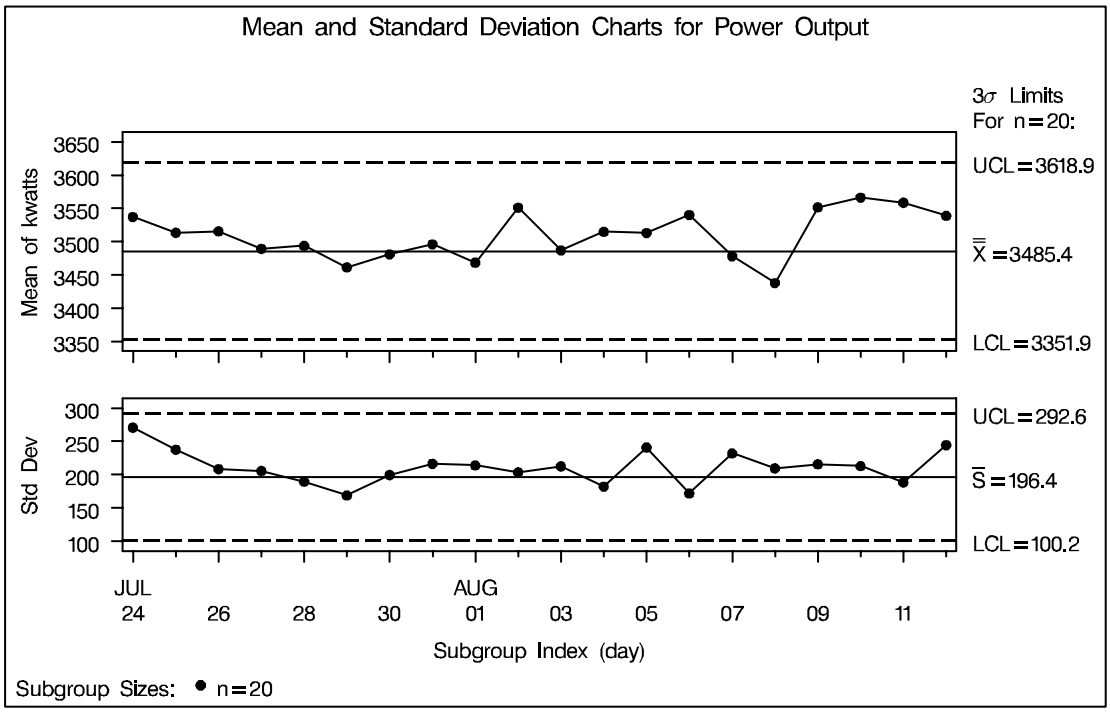


Figure 51.8. $\bar{X}$ and s Charts for Second Set of Power Outputs

The means and standard deviations lie within the control limits, indicating that the heating process is still in statistical control.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1820 for details concerning the variables that you must provide.

*In Release 6.09 and in earlier releases, it is also necessary to specify the READLIMITS option to read control limits from a LIMITS= data set.

Syntax

The basic syntax for the XSCHART statement is as follows:

XSCHART *process****** *subgroup-variable* **;**

The general form of this syntax is as follows:

XSCHART (*processes***)*** *subgroup-variable* <**(***block-variables* **)** >
< =*symbol-variable* | =’*character*’ > < **/** *options* >**;**

You can use any number of XSCHART statements in the SHEWHART procedure. The components of the XSCHART statement are described as follows.

process
processes
identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If the raw data are read using a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see “Creating Charts for Means and Standard Deviations from Raw Data” on page 1790.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see “Creating Charts for Means and Standard Deviations from Summary Data” on page 1793.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see “Saving Control Limits” on page 1797.

A *process* is required. If more than one *process* is specified, enclose the list in parentheses. For example, the following statements request distinct $\bar{X}$ and s charts for WEIGHT, LENGTH, and WIDTH:

```
proc shewhart data=measures;
   xschart (weight length width)*day;
run;
```

subgroup-variable
is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding XSCHART statement, DAY is the subgroup variable. For details, see “Subgroup Variables” on page 1771.

block-variables
are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See “Displaying Stratification in Blocks of Observations” on page 1932 for an example.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the means and standard deviations.

- If you produce a chart on a line printer, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce a chart on a graphics device, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOLn statements. See "Displaying Stratification in Levels of a Classification Variable" on page 1931 for an example.

character
specifies a plotting character for charts produced on line printers. For example, the following statements create $\bar{X}$ and s charts using an asterisk (*) to plot the points:

```
proc shewhart data=values;
   xschart weight*day='*';
run;
```

options
enhance the appearance of the charts, request additional analyses, save results in data sets, and so on. The "Summary of Options" section, which follows, lists all options by function. Chapter 53, "Dictionary of Options," describes each option in detail.

Summary of Options

The following tables list the XSCHART statement options by function. For complete descriptions, see Chapter 53, "Dictionary of Options."

Table 51.1. Tabulation Options

TABLE	creates a basic table of subgroup means, subgroup sample sizes, subgroup standard deviations, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with columns indicating which tests for special causes are positive

Note that specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.

Table 51.2. Options for Specifying Tests for Special Causes

NO3SIGMACHECK	allows tests to be applied with control limits other than 3σ limits
TESTS=*value-list*| *customized-pattern-list*	specifies tests for special causes for the $\bar{X}$ chart
TESTS2=*value-list*| *customized-pattern-list*	specifies tests for special causes for the s chart
TEST2RESET=*variable*	allows tests for special causes to be reset for the s chart
TEST2RUN=*n*	specifies length of pattern for Test 2
TEST3RUN=*n*	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL='*label*'| (*variable*)|*keyword*	provides labels for points where test is positive
TESTLABEL*n*='*label*'	specifies label for n^{th} test for special causes
TESTNMETHOD= STANDARDIZE	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
TESTRESET=*variable*	allows tests for special causes to be reset for the $\bar{X}$ chart
ZONELABELS	adds labels A, B, and C to zone lines for $\bar{X}$ chart
ZONE2LABELS	adds labels A, B, and C to zone lines for s chart
ZONES	adds lines to $\bar{X}$ chart delineating zones A, B, and C
ZONES2	adds lines to s chart delineating zones A, B, and C
ZONEVALPOS=*n*	specifies position of ZONEVALUES and ZONE2VALUES labels
ZONEVALUES	labels $\bar{X}$ chart zone lines with their values
ZONE2VALUES	labels s chart zone lines with their values

Table 51.3. Graphical Options for Displaying Tests for Special Causes

CTESTLABBOX=*color*	specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=*color*| *test-color-list*	specifies color for labels used to identify points where test is positive
CTESTSYMBOL=*color*	specifies color for symbol used to plot points where test is positive
CZONES=*color*	specifies color for lines and labels delineating zones A, B, and C
LTESTS=*linetype*	specifies type of line connecting points where test is positive
LZONES=*linetype*	specifies line type for lines delineating zones A, B, and C
TESTFONT=*font*	specifies software font for labels at points where test is positive
TESTHEIGHT=*value*	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=*symbol*	specifies plot symbol for points where test is positive
TESTSYMBOLHT=*value*	specifies symbol height for points where test is positive
WTESTS=*n*	specifies width of line connecting points where test is positive

Table 51.4. Line Printer Options for Displaying Tests for Special Causes

TESTCHAR='*character*'	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
ZONECHAR='*character*'	specifies character for lines that delineate zones for tests for special causes

Table 51.5. Reference Line Options

CHREF=*color*	specifies color for lines requested by the HREF= and HREF2= options
CVREF=*color*	specifies color for lines requested by the VREF= and VREF2= options
HREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on $\bar{X}$ chart
HREF2=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on s chart
HREFCHAR='*character*'	specifies line character for HREF= and HREF2= lines
HREFDATA= *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on $\bar{X}$ chart
HREF2DATA= *SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis on s chart
HREFLABELS= *'label1'...'labeln'*	specifies labels for HREF= lines
HREF2LABELS= *'label1'...'labeln'*	specifies labels for HREF2= lines
HREFLABPOS=*n*	specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=*linetype*	specifies line type for HREF= and HREF2= lines
LVREF=*linetype*	specifies line type for VREF= and VREF2= lines
NOBYREF	specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to vertical axis on $\bar{X}$ chart
VREF2=*values*| *SAS-data-set*	specifies position of reference lines perpendicular to vertical axis on s chart
VREFCHAR='*character*'	specifies line character for VREF= and VREF2= lines
VREFLABELS= *'label1'...'labeln'*	specifies labels for VREF= lines
VREF2LABELS= *'label1'...'labeln'*	specifies labels for VREF2= lines
VREFLABPOS=*n*	specifies position of VREFLABELS= and VREF2LABELS= labels

Table 51.6. Axis and Axis Label Options

CAXIS=*color*	specifies color for axis lines and tick marks
CFRAME=*color*|(*color-list*)	specifies fill colors for frame for plot area
CTEXT=*color*	specifies color for tick mark values and axis labels
HAXIS=*values*|AXISn	specifies major tick mark values for horizontal axis
HEIGHT=*value*	specifies height of axis label and axis legend text
HMINOR=*n*	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=*value*	specifies length of offset at both ends of horizontal axis
INTSTART=*value*	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	suppresses vertical axis truncation at zero applied by default to s chart
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=*n*	specifies thinning factor for tick mark labels on horizontal axis
SPLIT='*character*'	specifies splitting character for axis labels
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=*values*|AXISn	specifies major tick mark values for vertical axis of $\bar{X}$ chart
VAXIS2=*values*|AXISn	specifies major tick mark values for vertical axis of s chart
VFORMAT=*format*	specifies format for primary vertical axis tick mark labels
VFORMAT2=*format*	specifies format for secondary vertical axis tick mark labels
VMINOR=*n*	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=*value*	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis for primary chart
VZERO2	forces origin to be included in vertical axis for secondary chart
WAXIS=*n*	specifies width of axis lines

Table 51.7. Specification Limit Options

CIINDICES (ALPHA=*value* TYPE=*keyword*)	specifies α value and type for computing capability index confidence limits
LSL=*value-list*	specifies list of lower specification limits
TARGET=*value-list*	specifies list of target values
USL=*value-list*	specifies list of upper specification limits

Table 51.8. Clipping Options

CCLIP=*color*	specifies color for plot symbol for clipped points
CLIPCHAR='*character*'	specifies plot character for clipped points
CLIPFACTOR=*value*	determines extent to which extreme points are clipped
CLIPLEGEND='*string*'	specifies text for clipping legend
CLIPLEGPOS=*keyword*	specifies position of clipping legend
CLIPSUBCHAR='*character*'	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=*symbol*	specifies plot symbol for clipped points
CLIPSYMBOLHT=*value*	specifies symbol marker height for clipped points

Table 51.9. Block Variable Legend Options

BLOCKLABELPOS=*keyword*	specifies position of label for *block-variable* legend
BLOCKLABTYPE=*value*|*keyword*	specifies text size of *block-variable* legend
BLOCKPOS=*n*	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=*color-list*	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=*variable*|(*variables*)	specifies one or more variables whose values are colors for filling background of *block-variable* legend

Table 51.10. Options for Specifying Control Limits

ALPHA=*value*	requests probability limits for control charts
LIMITN=*n*|VARYING	specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	computes control limits for each *process* from the data rather than from a LIMITS= data set (Release 6.10 and later releases)
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEXES=ALL|'*label1*'...'*labeln*'	reads multiple sets of control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (Release 6.09 and earlier releases)
SIGMAS=*k*	specifies width of control limits in terms of multiple *k* of standard error of plotted means and standard deviations

Table 51.11. Options for Displaying Control Limits

CINFILL=*color*	specifies color for area inside control limits
CLIMITS=*color*	specifies color of control limits, central line, and related labels
LCLLABEL='*label*'	specifies label for lower control limit on $\bar{X}$ chart
LCLLABEL2='*label*'	specifies label for lower control limit on s chart
LIMLABSUBCHAR='*character*'	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=*linetype*	specifies line type for control limits
NDECIMAL=*n*	specifies number of digits to right of decimal place in default labels for control limits and central line on $\bar{X}$ chart
NDECIMAL2=*n*	specifies number of digits to right of decimal place in default labels for control limits and central line on s chart
NOCTL	suppresses display of central line on $\bar{X}$ chart
NOCTL2	suppresses display of central line on s chart
NOLCL	suppresses display of lower control limit on $\bar{X}$ chart
NOLCL2	suppresses display of lower control limit on s chart
NOLIMITLABEL	suppresses labels for control limits and central lines
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOLIMIT0	suppresses display of zero lower control limit on s chart
NOUCL	suppresses display of upper control limit on $\bar{X}$ chart
NOUCL2	suppresses display of upper control limit on s chart
SSYMBOL='*string*'|*keyword*	specifies label for central line on s chart
UCLLABEL='*string*'	specifies label for upper control limit on $\bar{X}$ chart
UCLLABEL2='*string*'	specifies label for upper control limit on s chart
WLIMITS=*n*	specifies width for control limits and central line
XSYMBOL='*string*'|*keyword*	specifies label for central line on $\bar{X}$ chart

Table 51.12. Grid Options

CGRID=*color*	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=*linetype*	specifies line type for grid requested with the ENDGRID option
LGRID=*linetype*	specifies line type for grid requested with the GRID option
WGRID=*n*	specifies width of grid lines

Table 51.13. Options for Plotting and Labeling Points

ALLLABEL=VALUE|(*variable*)	labels every point on $\bar{X}$ chart
ALLLABEL2=VALUE|(*variable*)	labels every point on s chart
CLABEL=*color*	specifies color for labels
CCONNECT=*color*	specifies color for line segments that connect points on chart
CFRAMELAB=*color*	specifies fill color for frame around labeled points
CNEEDLES=*color*	specifies color for needles that connect points to central line
CONNECTCHAR='*character*'	specifies character used to form line segments that connect points on chart
COUT=*color*	specifies color for portions of line segments that connect points outside control limits
COUTFILL=*color*	specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=*angle*	specifies angle at which labels are drawn
LABELFONT=*font*	specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=*value*	specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on chart
OUTLABEL=VALUE|(*variable*)	labels points outside control limits on $\bar{X}$ chart
OUTLABEL2=VALUE|(*variable*)	labels points outside control limits on s chart
SYMBOLCHARS='*characters*'	specifies characters indicating *symbol-variable*
SYMBOLLEGEND=NONE|*name*	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=*keyword*	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=*n*	specifies width of needles

Table 51.14. Input Data Set Options

MISSBREAK	specifies that observations with missing values are not to be processed

Table 51.15. Output Data Set Options

OUTHISTORY=*SAS-data-set*	creates output data set containing subgroup summary statistics
OUTINDEX='*string*'	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=*SAS-data-set*	creates output data set containing control limits
OUTTABLE=*SAS-data-set*	creates output data set containing subgroup summary statistics and control limits

Table 51.16. Phase Options

CPHASELEG=*color*	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE='*string*'	specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=*value*|*keyword*	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of charts
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=ALL|'*label1*'...'*labeln*'	specifies *phases* to be read from an input data set

Table 51.17. Process Mean and Standard Deviation Options

MU0=*value*	specifies known (standard) value of μ_0 for process mean μ
SIGMA0=*value*	specifies known (standard) value σ_0 for process standard deviation σ
SMETHOD=*keyword*	specifies method for estimating the process standard deviation σ
TYPE=*keyword*	identifies whether parameters are estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set

Table 51.18. Options for Interactive Control Charts

HTML=(*variable*)	specifies a variable whose values are URLs to be associated with subgroups
HTML2=(*variable*)	specifies variable whose values are URLs to be associated with subgroups on secondary chart
HTML_LEGEND=(*variable*)	specifies a variable whose values are URLs to be associated with symbols in the symbol legend
WEBOUT=*SAS-data-set*	creates an OUTTABLE= data set with additional graphics coordinate data

Table 51.19. Plot Layout Options

ALLN	plots means and standard deviations for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts for a process variable only when exceptions occur
INTERVAL=*keyword*	specifies natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=*n*	specifies maximum number of pages or screens for charts
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of charts
NOCHART2	suppresses creation of s chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=*n*	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
SEPARATE	displays $\bar{X}$ and s charts on separate screens or pages
TOTPANELS=*n*	specifies number of pages or screens to be used to display chart
YPCT1=*value*	specifies length of vertical axis on $\bar{X}$ chart as a percentage of the sum of lengths of vertical axes for $\bar{X}$ and s charts
ZEROSTD	displays $\bar{X}$ and s charts regardless of whether $\hat{\sigma} = 0$

Table 51.20. Graphical Enhancement Options

ANNOTATE=*SAS-data-set*	specifies annotate data set that adds features to $\bar{X}$ chart
ANNOTATE2=*SAS-data-set*	specifies annotate data set that adds features to s chart
DESCRIPTION='*string*'	specifies string that appears in the description field of the PROC GREPLAY master menu for $\bar{X}$ chart
DESCRIPTION2='*string*'	specifies string that appears in the description field of the PROC GREPLAY master menu for s chart
FONT=*font*	specifies software font for labels and legends on charts
NAME='*string*'	specifies name that appears in the name field of the PROC GREPLAY master menu for $\bar{X}$ chart
NAME2='*string*'	specifies name that appears in the name field of the PROC GREPLAY master menu for s chart
PAGENUM='*string*'	specifies the form of the label used in pagination
PAGENUMPOS= *keyword*	specifies the position of the page number requested with the PAGENUM= option

Table 51.21. Star Options

CSTARCIRCLES=*color*	specifies color for STARCIRCLES= circles
CSTARFILL=*color*| *(variable)*	specifies color for filling stars
CSTAROUT=*color*	specifies outline color for stars exceeding inner or outer circles
CSTARS=*color*| *(variable)*	specifies color for outlines of stars
LSTARCIRCLES= *linetypes*	specifies line types for STARCIRCLES= circles
LSTARS=*linetype*|*(variable)*	specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=*value*	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=*value-list*	specifies reference circles for stars
STARINRADIUS=*value*	specifies inner radius of stars
STARLABEL=*keyword*	specifies vertices to be labeled
STARLEGEND=*keyword*	specifies style of legend for star vertices
STARLEGENDLAB=*'label'*	specifies label for STARLEGEND= legend
STAROUTRADIUS=*value*	specifies outer radius of stars
STARSPEC=*value*| *SAS-data-set*	specifies method used to standardize vertex variables
STARSTART=*value*	specifies angle for first vertex
STARTYPE=*keyword*	specifies graphical style of star
STARVERTICES= *variable*|*(variables)*	superimposes star at each point on $\bar{X}$ chart
WSTARCIRCLES=*n*	specifies width of STARCIRCLES= circles
WSTARS=*n*	specifies width of STARVERTICES= stars

Table 51.22. Overlay Options

CCOVERLAY=*color-list*	specifies colors for primary chart overlay line segments
CCOVERLAY2=*color-list*	specifies colors for secondary chart overlay line segments
COVERLAY=*color-list*	specifies colors for primary chart overlay plots
COVERLAY2=*color-list*	specifies colors for secondary chart overlay plots
COVERLAYCLIP=*color*	specifies color for clipped points on overlays
LOVERLAY=*linetypes*	specifies line types for primary chart overlay line segments
LOVERLAY2=*linetypes*	specifies line types for secondary chart overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=*variable-list*	specifies variables to overlay on primary chart
OVERLAY2=*variable-list*	specifies variables to overlay on secondary chart
OVERLAY2HTML=*variable-list*	specifies URLS to associate with secondary chart overlay points
OVERLAY2ID=*variable-list*	specifies labels for secondary chart overlay points
OVERLAY2SYM=*symbol-list*	specifies symbols for secondary chart overlays
OVERLAY2SYMHT=*value-list*	specifies symbol heights for secondary chart overlays
OVERLAYCLIPSYM=*symbol*	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=*value*	specifies symbol height for clipped points on overlays
OVERLAYHTML=*variable-list*	specifies URLS to associate with primary chart overlay points
OVERLAYID=*variable-list*	specifies labels for primary chart overlay points
OVERLAYLEGLAB='*label*'	specifies label for overlay legend
OVERLAYSYM=*symbol-list*	specifies symbols for primary chart overlays
OVERLAYSYMHT=*value-list*	specifies symbol heights for primary chart overlays
WOVERLAY=*value-list*	specifies widths of primary chart overlay line segments
WOVERLAY2=*value-list*	specifies widths of secondary chart overlay line segments

Details

Constructing Charts for Means and Standard Deviations

The following notation is used in this section:

μ	process mean (expected value of the population of measurements)
σ	process standard deviation (standard deviation of the population of measurements)
$\bar{X}_i$	mean of measurements in i^{th} subgroup
s_i	standard deviation of the measurements $x_{i1}, \ldots, x_{ini}$ in the i^{th} subgroup $$s_i = \sqrt{((x_{i1} - \bar{X}_i)^2 + \cdots + (x_{in_i} - \bar{X}_i)^2)/(n_i - 1)}$$
n_i	sample size of i^{th} subgroup
N	number of subgroups
$\overline{\overline{X}}$	weighted average of subgroup means
z_p	$100p^{\text{th}}$ percentile of the standard normal distribution
$c_4(n)$	expected value of the standard deviation of n independent normally distributed variables with unit standard deviation
$c_5(n)$	standard error of the standard deviation of n independent observations from a normal population with unit standard deviation
$\chi^2_p(n)$	$100p^{\text{th}}$ percentile $(0 < p < 1)$ of the χ^2 distribution with n degrees of freedom

Plotted Points

Each point on an $\bar{X}$ chart indicates the value of a subgroup mean ($\bar{X}_i$). For example, if the tenth subgroup contains the values 12, 15, 19, 16, and 13, the mean plotted for this subgroup is

$$\bar{X}_{10} = \frac{12 + 15 + 19 + 16 + 13}{5} = 15$$

Each point on an s chart indicates the value of a subgroup standard deviation (s_i). For example, the standard deviation plotted for the tenth subgroup is

$$s_{10} = \sqrt{((12 - 15)^2 + (15 - 15)^2 + (19 - 15)^2 + (16 - 15)^2 + (13 - 15)^2)/4} = 2.739$$

Central Lines

On an $\bar{X}$ chart, by default, the central line indicates an estimate of μ, which is computed as

$$\hat{\mu} = \overline{\overline{X}} = \frac{n_1\bar{X}_1 + \cdots + n_N\bar{X}_N}{n_1 + \cdots + n_N}$$

If you specify a known value (μ_0) for μ, the central line indicates the value of μ_0.

On the s chart, by default, the central line for the i^{th} subgroup indicates an estimate for the expected value of s_i, which is computed as $c_4(n_i)\hat{\sigma}$, where $\hat{\sigma}$ is an estimate of σ. If you specify a known value (σ_0) for σ, the central line indicates the value of $c_4(n_i)\sigma_0$. Note that the central line varies with n_i.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard errors of $\bar{X}_i$ and s_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that $\bar{X}_i$ or s_i exceeds the limits

The following table provides the formulas for the limits:

Table 51.23. Limits for $\bar{X}$ and s Charts

Control Limits	
$\bar{X}$ Chart	LCL = lower limit = $\overline{\overline{X}} - k\hat{\sigma}/\sqrt{n_i}$ UCL = upper limit = $\overline{\overline{X}} + k\hat{\sigma}/\sqrt{n_i}$
s Chart	LCL = lower limit = $\max(c_4(n_i)\hat{\sigma} - kc_5(n_i)\hat{\sigma}, 0)$ UCL = upper limit = $c_4(n_i)\hat{\sigma} + kc_5(n_i)\hat{\sigma}$

Probability Limits	
$\bar{X}$ Chart	LCL = lower limit = $\overline{\overline{X}} - z_{\alpha/2}(\hat{\sigma}/\sqrt{n_i})$ UCL = upper limit = $\overline{\overline{X}} + z_{\alpha/2}(\hat{\sigma}/\sqrt{n_i})$
s Chart	LCL = lower limit = $\hat{\sigma}\sqrt{\chi^2_{\alpha/2}(n_i - 1)/(n_i - 1)}$ UCL = upper limit = $\hat{\sigma}\sqrt{\chi^2_{1-\alpha/2}(n_i - 1)/(n_i - 1)}$

The formulas for s charts assume that the data are normally distributed. If standard values μ_0 and σ_0 are available for μ and σ, respectively, replace $\overline{\overline{X}}$ with μ_0 and $\hat{\sigma}$ with σ_0 in Table 51.23. Note that the limits vary with n_i and that the probability limits for s_i are asymmetric about the central line.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify μ_0 with the MU0= option or with the variable _MEAN_ in a LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables are saved:

Table 51.24. OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
_CP_	capability index C_p
CPK	capability index C_{pk}
CPL	capability index CPL
CPM	capability index C_{pm}
CPU	capability index CPU
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLS	lower control limit for subgroup standard deviation
LCLX	lower control limit for subgroup mean
LIMITN	nominal sample size associated with the control limits
LSL	lower specification limit
MEAN	process mean ($\overline{\overline{X}}$ or μ_0)
_S_	value of central line on s chart
SIGMAS	multiple (k) of standard error of $\bar{X}_i$ or s_i
STDDEV	process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*subgroup-variable* specified in the XSCHART statement
TARGET	target value
TYPE	type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLS	upper control limit for subgroup standard deviation
UCLX	upper control limit for subgroup mean
USL	upper specification limit
VAR	*process* specified in the XSCHART statement

Notes:

1. If the control limits vary with subgroup sample size, the special missing value *V* is assigned to the variables _LIMITN_, _LCLX_, _UCLX_, _LCLS_, _S_, and _UCLS_.
2. If the limits are defined in terms of a multiple k of the standard errors of $\bar{X}_i$ and s_i, the value of _ALPHA_ is computed as $\alpha = 2(1 - \Phi(k))$, where $\Phi(\cdot)$ is the standard normal distribution function.
3. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = \Phi^{-1}(1 - \alpha/2)$, where Φ^{-1} is the inverse standard normal distribution function.
4. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you

provide a target value with the TARGET= option. See "Capability Indices" on page 1774 for computational details.

5. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the XSCHART statement. For an example, see "Saving Control Limits" on page 1797.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables are saved:

- the *subgroup-variable*
- a subgroup mean variable named by *process* suffixed with *X*
- a subgroup standard deviation variable named by *process* suffixed with *S*
- a subgroup sample size variable named by *process* suffixed with *N*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the XSCHART statement. For example, consider the following statements:

```
proc shewhart data=steel;
   xschart (width diameter)*lot / outhistory=summary;
run;
```

The data set SUMMARY contains variables named LOT, WIDTHX, WIDTHS, WIDTHN, DIAMTERX, DIAMTERS, and DIAMTERN. Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 1796.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables are saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on $\bar{X}$ chart
EXLIMS	control limit exceeded on s chart
LCLS	lower control limit for standard deviation
LCLX	lower control limit for mean
LIMITN	nominal sample size associated with the control limits
MEAN	process mean
_S_	average standard deviation
SIGMAS	multiple (k) of the standard error associated with control limits
subgroup	values of the subgroup variable
SUBN	subgroup sample size
SUBS	subgroup standard deviation
SUBX	subgroup mean
TESTS	tests for special causes signaled on $\bar{X}$ chart
TESTS2	tests for special causes signaled on s chart
UCLS	upper control limit for standard deviation
UCLX	upper control limit for mean
VAR	*process* specified in the XSCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The k^{th} character of a value of _TESTS_ is k if Test k is positive at that subgroup. For example, if you request all eight tests and Tests 2 and 8 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, an 8 for the eighth character, and blanks for the other six characters.
3. The variable _TESTS2_ is saved if you specify the TESTS2= option.
4. The variables _EXLIM_, _EXLIMS_, _TESTS_, and _TESTS2_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1797.

ODS Tables

The following table summarizes the ODS tables that you can request with the XSCHART statement.

Table 51.25. ODS Tables Produced with the XSCHART Statement

Table Name	Description	Options
XSCHART	$\bar{X}$ and s chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the XSCHART statement must be a SAS variable in the DATA= data set. This variable provides measurements that must be grouped into subgroup samples indexed by the *subgroup-variable*. The *subgroup-variable*, which is specified in the XSCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. If the i^{th} subgroup contains n_i items, there should be n_i consecutive observations for which the value of the subgroup variable is the index of the i^{th} subgroup. For example, if each subgroup contains five items and there are 30 subgroup samples, the DATA= data set should contain 150 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all the observations in a DATA= data set. However, if the DATA= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 1936).

For an example of a DATA= data set, see "Creating Charts for Means and Standard Deviations from Raw Data" on page 1790.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set CONLIMS:*

```
proc shewhart data=info limits=conlims;
   xschart weight*batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLX_, _MEAN_, _UCLX_, _LCLS_, _S_, and _UCLS_, which specify the control limits directly
- the variables _MEAN_ and _STDDEV_, which are used to calculate the control limits according to the equations in Table 51.23 on page 1815

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are `ESTIMATE`, `STANDARD`, `STDMU`, and `STDSIGMA`.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1800.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This allows you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART, CUSUM, or MACONTROL procedures or to read output data sets created with SAS summarization procedures, such as PROC MEANS.

*In Release 6.09 and in earlier releases, it is also necessary to specify the READLIMITS option.

A HISTORY= data set used with the XSCHART statement must contain the following variables:

- the *subgroup-variable*
- a subgroup mean variable for each *process*
- a subgroup standard deviation variable for each *process*
- a subgroup sample size variable for each *process*

The names of the subgroup mean, subgroup standard deviation, and subgroup sample size variables must be the *process* name concatenated with the special suffix characters X, S, and N, respectively. For example, consider the following statements:

```
proc shewhart history=summary;
   xschart (weight yldstren)*batch;
run;
```

The data set SUMMARY must include the variables BATCH, WEIGHTX, WEIGHTS, WEIGHTN, YLDSRENX, YLDSRENS, and YLDSRENN.

Note that if you specify a *process* name that contains 32 characters, the names of summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 1936 for an example).

For an example of a HISTORY= data set, see "Creating Charts for Means and Standard Deviations from Summary Data" on page 1793.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information read from a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in Chapter 56, "Specialized Control Charts."

The following table lists the variables required in a TABLE= data set used with the XSCHART statement:

Table 51.26. Variables Required in a TABLE= Data Set

Variable	Description
LCLS	lower control limit for standard deviation
LCLX	lower control limit for mean
LIMITN	nominal sample size associated with the control limits
MEAN	process mean
_S_	average standard deviation
subgroup-variable	values of the *subgroup-variable*
SUBN	subgroup sample size
SUBS	subgroup standard deviation
SUBX	subgroup mean
UCLS	upper control limit for standard deviation
UCLX	upper control limit for mean

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes for subgroup means and must be a character variable of length 8.
- _TESTS2_ (if the TESTS2= option is specified). This variable is used to flag tests for special causes for subgroup standard deviations and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1797.

Methods for Estimating the Standard Deviation

When control limits are determined from the input data, four methods (referred to as default, MVLUE, MVGRANGE, and RMSDF) are available for estimating σ.

Default Method

The default estimate for σ is

$$\hat{\sigma} = \frac{s_1/c_4(n_1) + \cdots + s_N/c_4(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, s_i is the sample standard deviation of the i^{th} subgroup

$$s_i = \sqrt{\frac{1}{n_i - 1}\sum_{j=1}^{n_i}(x_{ij} - \bar{X}_i)^2}$$

and

$$c_4(n_i) = \frac{\Gamma(n_i/2)\sqrt{2/(n_i - 1)}}{\Gamma((n_i - 1)/2)}$$

Here, $\Gamma(\cdot)$ denotes the gamma function, and $\bar{X}_i$ denotes the i^{th} subgroup mean. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$. If the observations are normally distributed, then the expected value of s_i is $c_4(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the *ASTM Manual on Presentation of Data and Control Chart Analysis* (1976).

MVLUE Method

If you specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). This estimate is a weighted average of N unbiased estimates of σ of the form $s_i/c_4(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{h_1 s_1/c_4(n_1) + \cdots + h_N s_N/c_4(n_N)}{h_1 + \cdots + h_N}$$

where

$$h_i = \frac{[c_4(n_i)]^2}{1 - [c_4(n_i)]^2}$$

A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

MVGRANGE Method

If you specify SMETHOD=MVGRANGE, σ is estimated using a moving range of subgroup averages. This is appropriate for constructing control charts for means when the jth measurement in the ith subgroup can be modeled as $x_{ij} = \sigma_B \omega_i + \sigma_W \epsilon_{ij}$, where σ_B^2 is the between-subgroup variance, σ_W^2 is the within-subgroup variance, the ω_i are independent with zero mean and unit variance, and the ω_i are independent of the ϵ_{ij}.

The estimate for σ is

$$\hat{\sigma} = \bar{R}/d_2(n)$$

where $\bar{R}$ is the average of the moving ranges, n is the number of consecutive subgroup averages used to compute each moving range, and the unbiasing factor $d_2(n)$ is defined so that if the subgroup averages are normally distributed, the expected value of R_i is

$$E(R_i) = d_2(n_i)\sigma$$

This method is appropriate for constructing a variation on the three-way control chart that is advocated for this situation by Wheeler (1995). A three-way control chart is useful when sampling, or *within-group* variation is not the only source of variation, as discussed in "Multiple Components of Variation" on page 2009. Wheeler's three-way control chart comprises a chart of subgroup means, a moving range chart of the subgroup means, and a chart of subgroup ranges. This variation substitutes a chart of subgroup standard deviations for the chart of subgroup ranges. When you specify the SMETHOD=MVGRANGE option, the XSCHART statement produces the appropriate charts of subgroup means and subgroup standard deviations.

RMSDF Method

If you specify SMETHOD=RMSDF, a weighted root-mean-square estimate is computed for σ.

$$\hat{\sigma} = \frac{\sqrt{(n_1 - 1)s_1^2 + \cdots + (n_N - 1)s_N^2}}{c_4(n)\sqrt{n_1 + \cdots + n_N - N}}$$

The weights are the degrees of freedom $n_i - 1$. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$.

If the unknown standard deviation σ is constant across subgroups, the root-mean-square estimate is more efficient than the minimum variance linear unbiased estimate. However, in process control applications it is generally not assumed that σ is constant, and if σ varies across subgroups, the root-mean-square estimate tends to be more inflated than the MVLUE.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical ($\bar{X}$ chart)	DATA=	*process*
Vertical ($\bar{X}$ chart)	HISTORY=	subgroup mean variable
Vertical ($\bar{X}$ chart)	TABLE=	_SUBX_

You can specify distinct labels for the vertical axes of the $\bar{X}$ and s charts by breaking the vertical axis into two parts with a split character. Specify the split character with the SPLIT= option. The first part labels the vertical axis of the $\bar{X}$ chart, and the second part labels the vertical axis of the s chart.

For example, the following sets of statements specify the label *Avg Power Output* for the vertical axis of the $\bar{X}$ chart and the label *Std Deviation* for the vertical axis of the s chart:

```
proc shewhart data=turbine;
   xschart kwatts*day / split = '/' ;
   label kwatts = 'Avg Power Output/Std Deviation';
run;

proc shewhart history=turbhist;
   xschart kwatts*day / split = '/' ;
   label kwattsx = 'Avg Power Output/Std Deviation';
run;

proc shewhart table=turbtab;
   xschart kwatts*day / split = '/' ;
   label _SUBX_ = 'Avg Power Output/Std Deviation';
run;
```

In this example, the label assignments are in effect only for the duration of the procedure step, and they temporarily override any permanent labels associated with the variables.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples

This section provides advanced examples of the XSCHART statement.

Example 51.1. Specifying Probability Limits

See SHWXS2 in the SAS/QC Sample Library

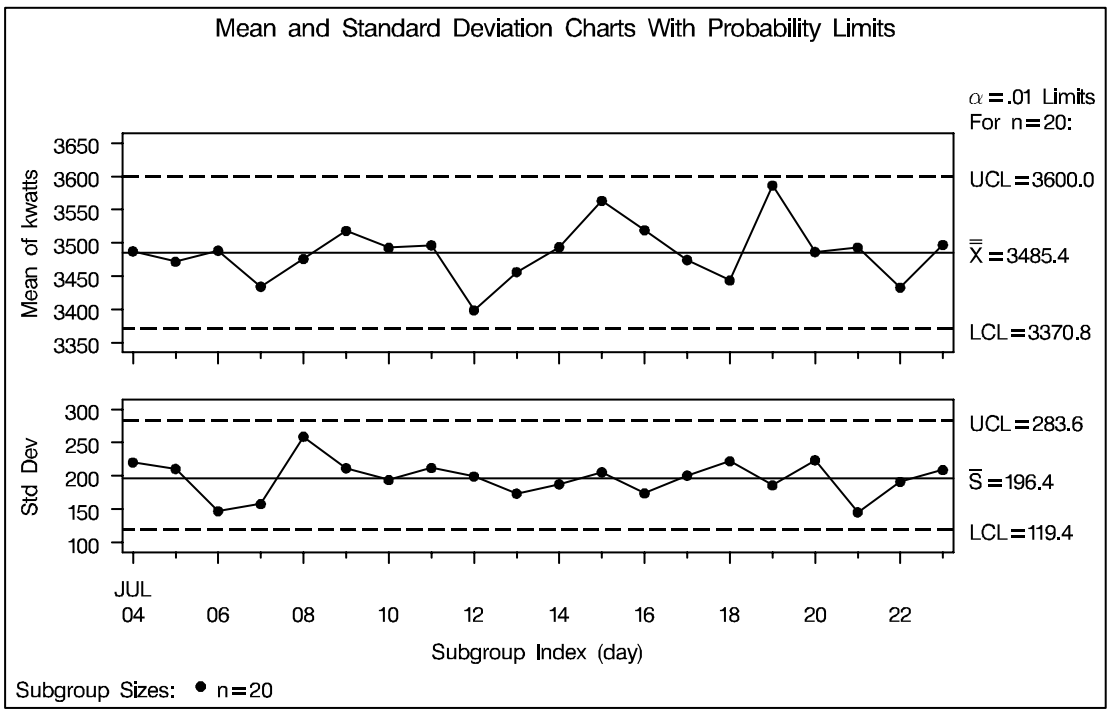
This example illustrates how to create $\bar{X}$ and s charts with probability limits. The following statements read the kilowatt power output measurements from the data set TURBINE (see page 1790) and create the $\bar{X}$ and s charts shown in Output 51.1.1:

```
title 'Mean and Standard Deviation Charts With Probability Limits';
proc shewhart data=turbine;
   xschart kwatts*day  / alpha     = 0.01
                         outlimits = oillim;
run;
```

The ALPHA= option specifies the probability (α) that a subgroup summary statistic is outside the limits. Here, the limits are computed so that the probability that a subgroup mean or standard deviation is less than its lower limit is $\alpha/2 = 0.005$, and the probability that a subgroup mean or standard deviation is greater than its upper limit is $\alpha/2 = 0.005$. This assumes that the measurements are normally distributed.

The OUTLIMITS= option names an output data set (OILSUM) that saves the probability limits. The data set OILLIM is shown in Output 51.1.2.

Output 51.1.1. Probability Limits on $\bar{X}$ and s Charts



Output 51.1.2. Probability Limit Information

Mean and Standard Deviation Charts with Probability Limits

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_
kwatts	day	ESTIMATE	20	0.01	2.57583	3370.79

MEAN	_UCLX_	_LCLS_	_S_	_UCLS_	_STDDEV_
3485.41	3600.03	119.432	196.396	283.570	198.996

The variable _ALPHA_ saves the value of α. The value of the variable _SIGMAS_ is computed as $k = \Phi^{-1}(1 - \alpha/2)$, where Φ^{-1} is the inverse standard normal distribution function. Note that, in this case, the probability limits for the mean are equivalent to 2.58σ limits.

Since all the points fall within the probability limits, it can be concluded that the process is in statistical control.

Example 51.2. Computing Subgroup Summary Statistics

See SHWXS3 in the SAS/QC Sample Library

You can use output data sets from a number of SAS procedures as input data sets for the SHEWHART procedure. In this example, the MEANS procedure is used to create a data set containing subgroup summary statistics, which can be read by the SHEWHART procedure as a HISTORY= data set. The following statements create an output data set named OILSUMM, which contains subgroup means, standard deviations, and sample sizes for the variable KWATTS in the data set TURBINE (see page 1790):

```
proc means data=turbine noprint;
  var kwatts;
  by day;
  output out=oilsumm mean=means std=stds n=sizes;
run;
```

A listing of OILSUMM is shown in Output 51.2.1.

The variables MEANS, STDS, and SIZES do not follow the naming convention required for HISTORY= data sets (see "HISTORY= Data Set" on page 1820). The following statements temporarily rename these variables to KWATTSX, KWATTSS, and KWATTSN, respectively (the names required when the *process* KWATTS is specified in the XSCHART statement):

```
title 'Mean and Standard Deviation Charts for Power Output';
symbol v=dot;
proc shewhart
   history=oilsumm (rename=(means  = kwattsx
                            stds   = kwattss
                            sizes  = kwattsn ));
  xschart kwatts*day;
run;
```

The resulting charts are identical to the charts in Figure 51.2 on page 1792.

Output 51.2.1. The Data Set OILSUMM

Summary Statistics for Power Output Data

day	kwattsx	kwattss	kwattsn
04JUL	3487.40	220.260	20
05JUL	3471.65	210.427	20
06JUL	3488.30	147.025	20
07JUL	3434.20	157.637	20
08JUL	3475.80	258.949	20
09JUL	3518.10	211.566	20
10JUL	3492.65	193.779	20
11JUL	3496.40	212.024	20
12JUL	3398.50	199.201	20
13JUL	3456.05	173.455	20
14JUL	3493.60	187.465	20
15JUL	3563.30	205.472	20
16JUL	3519.05	173.676	20
17JUL	3474.20	200.576	20
18JUL	3443.60	222.084	20
19JUL	3586.35	185.724	20
20JUL	3486.45	223.474	20
21JUL	3492.90	145.267	20
22JUL	3432.80	190.994	20
23JUL	3496.90	208.858	20

Example 51.3. Analyzing Nonnormal Process Data

See SHWXS4 in the SAS/QC Sample Library

The standard control limits for s charts (see Table 51.23 on page 1815) are calculated under the assumption that the data are normally distributed. This example illustrates how a transformation to normality can be used in conjunction with $\bar{X}$ and s charts.

The length of a metal brace is measured in centimeters for each of 20 braces sampled daily. Subgroup samples are collected for nineteen days, and the data are analyzed to determine if the manufacturing process is in statistical control.

```
data lengdata;
   informat day date7.;
   format day date5.;
   label length='Brace Length in cm';
   input day @;
   do i=1 to 5;
      input length @;
      output;
      end;
   drop i;
   datalines;
02JAN86  113.64  119.60  111.66  111.88  125.29
02JAN86  114.08  115.28  127.84  109.97  109.34
02JAN86  109.65  121.76  112.17  116.01  111.64
02JAN86  112.70  114.43  110.27  114.76  125.89
03JAN86  115.92  113.62  117.52  114.44  118.08
03JAN86  111.13  118.42  112.16  112.25  107.71
03JAN86  110.46  113.78  109.89  114.59  116.98
03JAN86  111.06  113.76  115.53  110.88  115.47
```

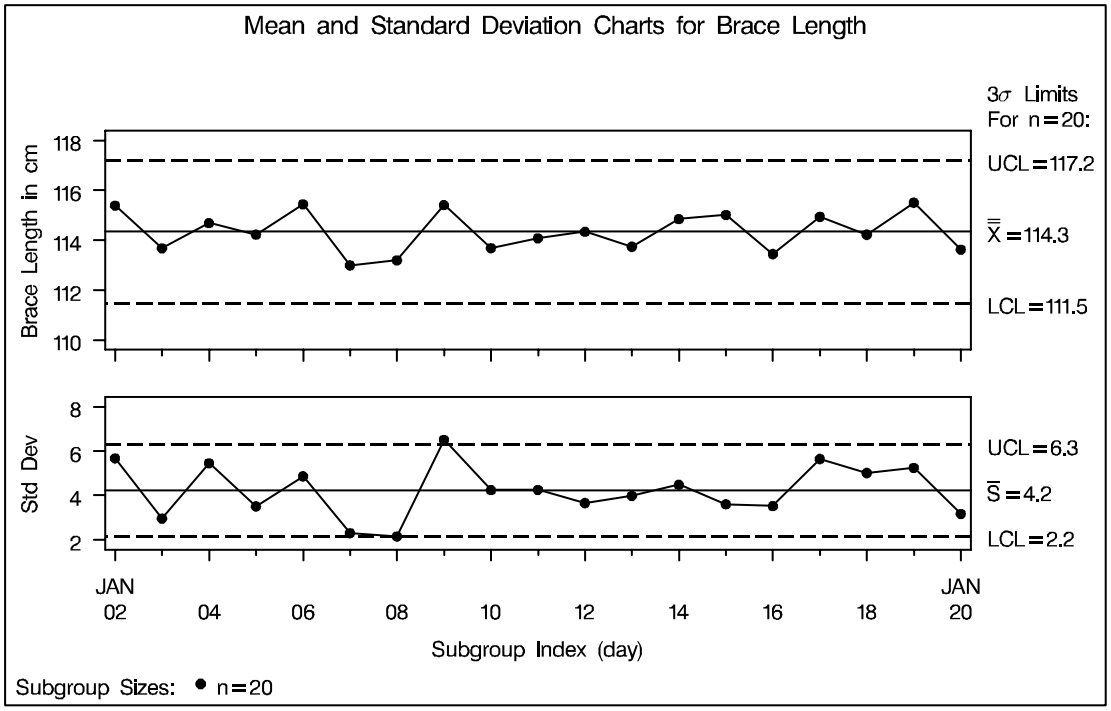
```
   ...

20JAN86  115.15  112.34  114.99  109.70  111.20
20JAN86  117.81  119.51  109.03  111.61  118.01
20JAN86  113.55  114.78  112.91  111.87  118.54
;
run;
```

The following statements create preliminary $\bar{X}$ and s charts for the lengths:

```
title 'Mean and Standard Deviation Charts for Brace Length';
proc shewhart data=lengdata;
   xschart length*day;
run;
```

The charts are shown in Output 51.3.1.

Output 51.3.1. $\bar{X}$ and s Charts



The s chart suggests that the process is not in control, since the standard deviation of the measurements recorded on January 9 exceeds its upper control limit. In addition, a number of other points on the s chart are close to the control limits.

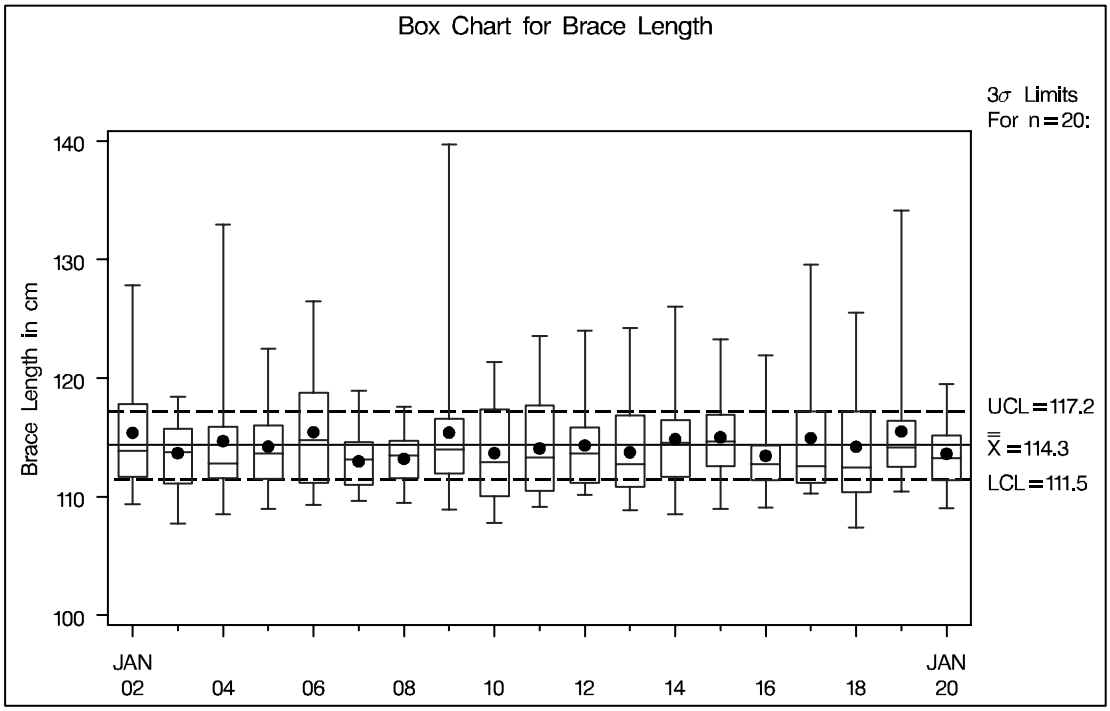
The following statements create a box chart for the lengths (for more information on box charts, see Chapter 39, "BOXCHART Statement,").

```
title 'Box Chart for Brace Length';
proc shewhart data=lengdata;
   boxchart length*day / serifs
                         ranges
                         nohlabel
                         nolegend;
run;
```

The chart, shown in Output 51.3.2, reveals that most of the subgroup distributions are skewed to the right. Consequently, the s chart shown in Output 51.3.1 should be interpreted with caution, since control limits for s charts are based on the assumption that the data are normally distributed.

No special cause for the skewness of the subgroup distributions is discovered. This indicates that the process is in statistical control and that the length distribution is naturally skewed.

Output 51.3.2. Box Chart



The following statements apply a lognormal transformation to the length measurements and display a box chart for the transformed data:

```
data lengdata;
   set lengdata;
   logleng=log(length-105);
label logleng='log of Length minus 105';
run;
```

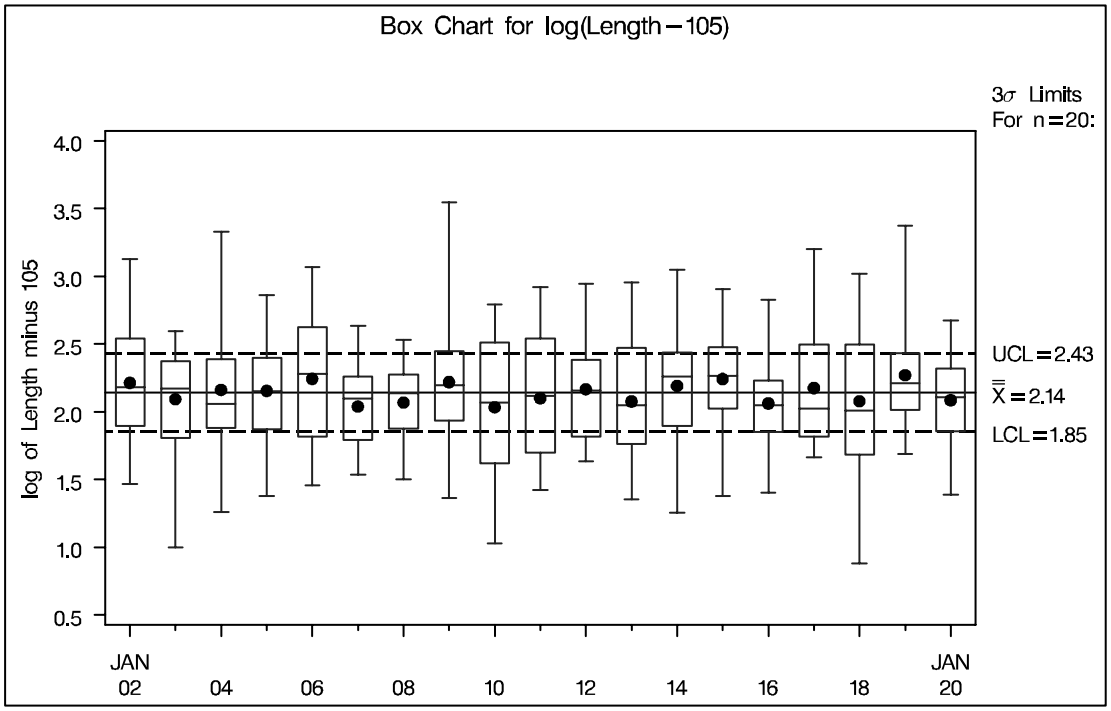
```
title 'Box Chart for log(Length-105)';
proc shewhart data=lengdata;
   boxchart logleng*day / serifs
                          ranges
                          nohlabel
                          nolegend;
run;
```

The chart, shown in Output 51.3.3, indicates that the subgroup distributions of LOGLENG are approximately normal (this can be verified with goodness-of-fit tests by using the CAPABILITY procedure).

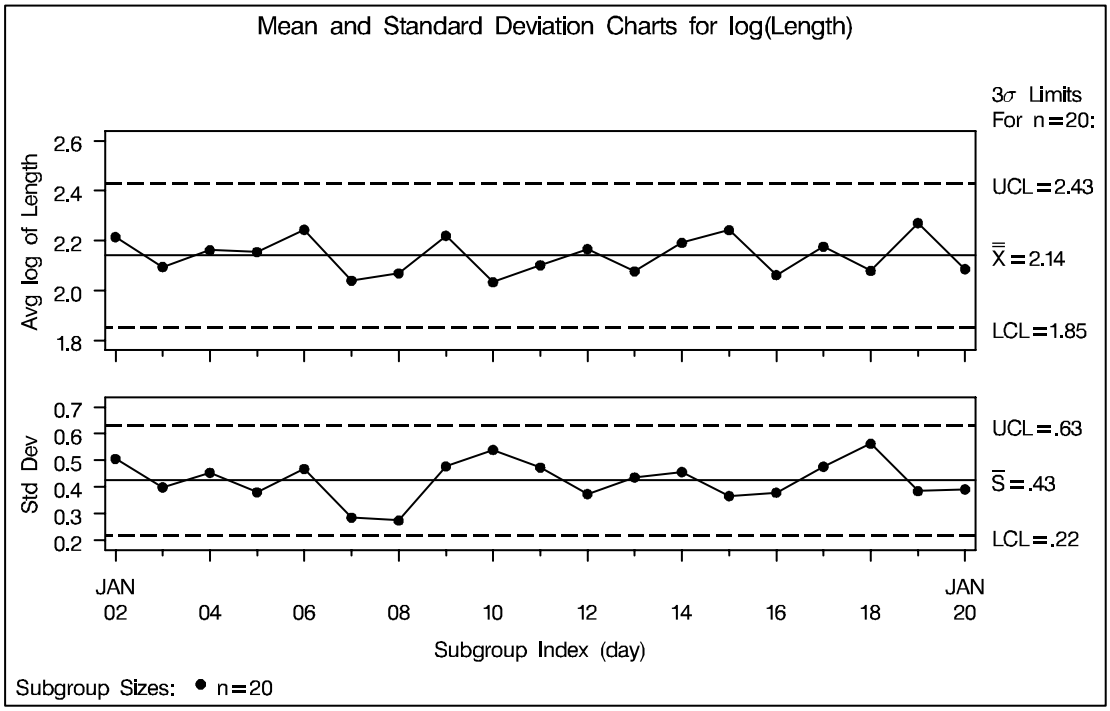
Finally, $\bar{X}$ and s charts, shown in Output 51.3.4, are created for LOGLENG. They indicate that the variability and mean level of the transformed lengths are in control.

```
title 'Mean and Standard Deviation Charts for log(Length)';
proc shewhart data=lengdata;
   xschart logleng*day / split = '/';
   label logleng='Avg log of Length/Std Dev';
run;
```

Output 51.3.3. Box Chart for Transformed Data



Output 51.3.4. $\bar{X}$ and s Charts for Transformed Length



Chapter 52
INSET and INSET2 Statements

Chapter Contents

OVERVIEW 1835

GETTING STARTED 1836
Displaying Summary Statistics on a Control Chart 1836
Formatting Values and Customizing Labels 1838
Adding a Header and Positioning the Inset 1839

SYNTAX 1841
Summary of INSET Keywords 1842
Summary of Options 1844
Dictionary of Options 1845

DETAILS 1847
Positioning the Inset Using Compass Points 1847
Positioning the Inset in the Margins 1848
Positioning the Inset Using Coordinates 1848

Chapter 52
INSET and INSET2 Statements

Overview

The INSET and INSET2 statements enable you to enhance a Shewhart chart by adding a box or table (referred to as an *inset*) of summary statistics directly to the graph. The INSET statement places an inset in a primary Shewhart chart while the INSET2 statement places one in a secondary Shewhart chart. An inset can display statistics calculated by the SHEWHART procedure or arbitrary values provided in a SAS data set.

Note that an INSET or INSET2 statement by itself does not produce a display but must be used in conjunction with a chart statement. Insets are not available with line printer output, so the INSET and INSET2 statements are not applicable when the LINEPRINTER option is specified in the PROC SHEWHART statement.

You can use options in the INSET and INSET2 statements to

- specify the position of the inset
- specify a header for the inset table
- specify graphical enhancements, such as background colors, text colors, text height, text font, and drop shadows

The INSET2 statement differs from the INSET statement in only two respects.

1. An INSET2 statement creates an inset within a secondary chart generated by an IRCHART, MRCHART, XRCHART or XSCHART statement or by the TRENDVAR= option. For example, when following an XRCHART statement an INSET statement produces an inset in the $\bar{X}$ chart and an INSET2 statement produces one in the R chart.
2. The INSET statement can be used to place an inset in one of the margins surrounding the plot area, while the INSET2 statement cannot.

Any of the statistics available for display in an inset can be specified with either an INSET or INSET2 statement. Descriptions of the INSET statement in this chapter also apply to the INSET2 statement except where explicitly noted.

Getting Started

This section introduces the INSET statement with examples that illustrate commonly used options. Complete syntax for the INSET statement is presented in the "Syntax" section on page 1841.

Displaying Summary Statistics on a Control Chart

In the manufacture of silicon wafers, batches of five wafers are sampled, and their diameters are measured in millimeters. The following statements create a SAS data set named WAFERS, which contains the measurements for 25 batches:

```
data wafers;
   input batch @;
   do i=1 to 5;
      input diamtr @;
      output;
      end;
   drop i;
datalines;
 1  35.00 34.99 34.99 34.98 35.00
 2  35.01 34.99 34.99 34.98 35.00
 3  34.99 35.00 35.00 35.00 35.00
 4  35.01 35.00 34.99 34.99 35.00
 5  35.00 34.99 34.98 34.99 35.00
 6  34.99 34.99 35.00 35.00 35.00
 7  35.01 34.98 35.00 35.00 34.99
 8  35.00 35.00 34.99 34.98 34.99
 9  34.99 34.98 34.98 35.01 35.00
10  34.99 35.00 35.01 34.99 35.01
11  35.01 35.00 35.00 34.98 34.99
12  34.99 34.99 35.00 34.98 35.01
13  35.01 34.99 34.98 34.99 34.99
14  35.00 35.00 34.99 35.01 34.99
15  34.98 34.99 34.99 34.98 35.00
16  34.99 35.00 35.00 35.01 35.00
17  34.98 34.98 34.99 34.99 34.98
18  35.01 35.02 35.00 34.98 35.00
19  34.99 34.98 35.00 34.99 34.98
20  34.99 35.00 35.00 34.99 34.99
21  35.00 34.99 34.99 34.98 35.00
22  35.00 35.00 35.01 35.00 35.00
23  35.02 35.00 34.98 35.02 35.00
24  35.00 35.00 34.99 35.01 34.98
25  34.99 34.99 34.99 35.00 35.00
;
run;
```

The following statements generate an $\bar{X}$ chart from the WAFERS data. Lower and upper specification limits for wafer diameters are given and the process capability index C_p is computed. An INSET statement is used to display the specification limits, the computed value of C_p and the process standard deviation on the chart:

```
title 'Mean Chart for Diameters';
proc shewhart data=wafers;
   xchart diamtr*batch /
      lsl = 34.97
      usl = 35.03;
   inset lsl usl cp stddev /
      height = 3;
run;
```

The resulting $\bar{X}$ chart is displayed in Figure 52.1. The INSET statement immediately follows the chart statement that creates the graphical display (in this case, the XCHART statement). Specify the keywords for inset statistics (such as LSL, USL, CP and STDDEV) immediately after the word INSET. The inset statistics appear in the order in which you specify the keywords. The HEIGHT= option on the INSET statement specifies the text height used to display the statistics in the inset.

A complete list of keywords that you can use with the INSET statement is provided in "Summary of INSET Keywords" on page 1842. Note that the set of keywords available for a particular display depends on both the plot statement that precedes the INSET statement and the options that you specify in the plot statement.

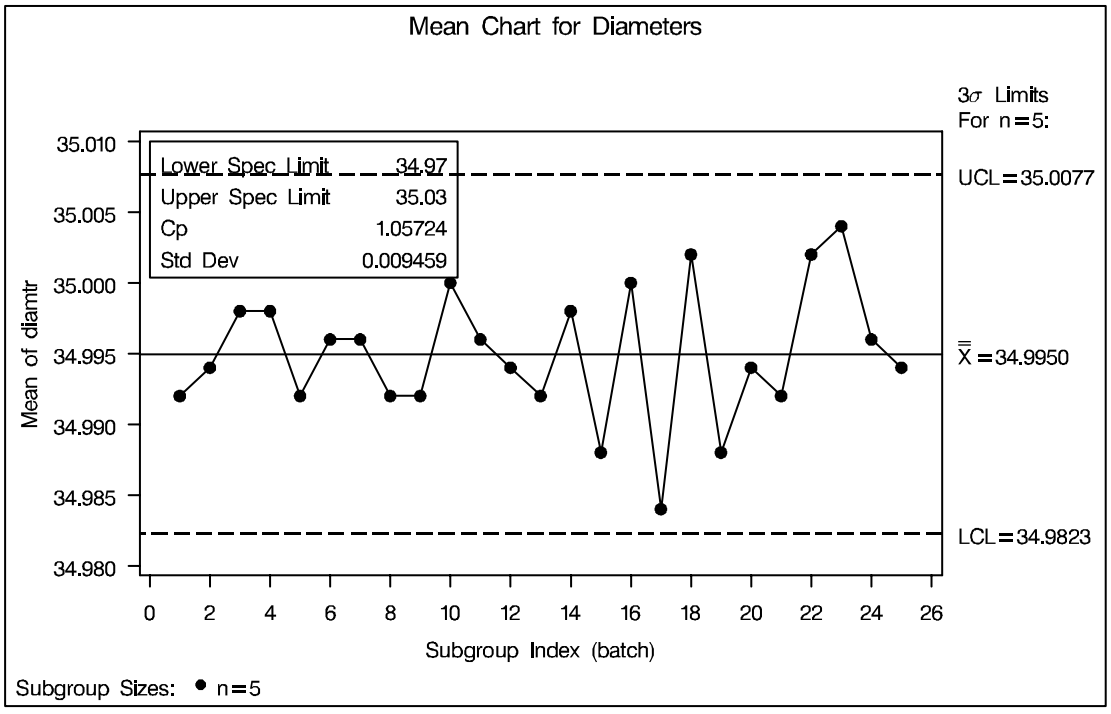


Figure 52.1. An $\bar{X}$ Chart with an Inset

The following examples illustrate options commonly used for enhancing the appearance of an inset.

Formatting Values and Customizing Labels

By default, each inset statistic is identified with an appropriate label, and each numeric value is printed using an appropriate format. However, you may want to provide your own labels and formats. For example, in Figure 52.1 the default format used for C_p and the process standard deviation prints an excessive number of decimal places. The following statements produce $\bar{X}$ and R charts, each with its own inset. The unwanted decimal places are eliminated and the default specification limits labels are replaced with abbreviations:

```
title 'Mean Chart for Diameters';
proc shewhart data=wafers;
   xrchart diamtr*batch /
      lsl = 34.97
      usl = 35.03;
   inset lsl='LSL' usl='USL' /
      pos = nw
      height = 3;
   inset2 cp (6.4) stddev (6.4) /
      pos = nw
      height = 3;
run;
```

The resulting $\bar{X}$ and R charts are displayed in Figure 52.2. You can provide your own label by specifying the keyword for that statistic followed by an equal sign (=) and the label in quotes. The label can have up to 24 characters.

The format 6.4 specified in parentheses after the CP and STDDEV keywords displays those statistics with a field width of six and four decimal places. In general, you can specify any numeric SAS format in parentheses after an inset keyword. You can also specify a format to be used for all the statistics in the INSET statement with the FORMAT= option. For more information about SAS formats, refer to Chapter 14 of *SAS Language Reference: Dictionary*.

Note that if you specify both a label and a format for a statistic, the label must appear before the format.

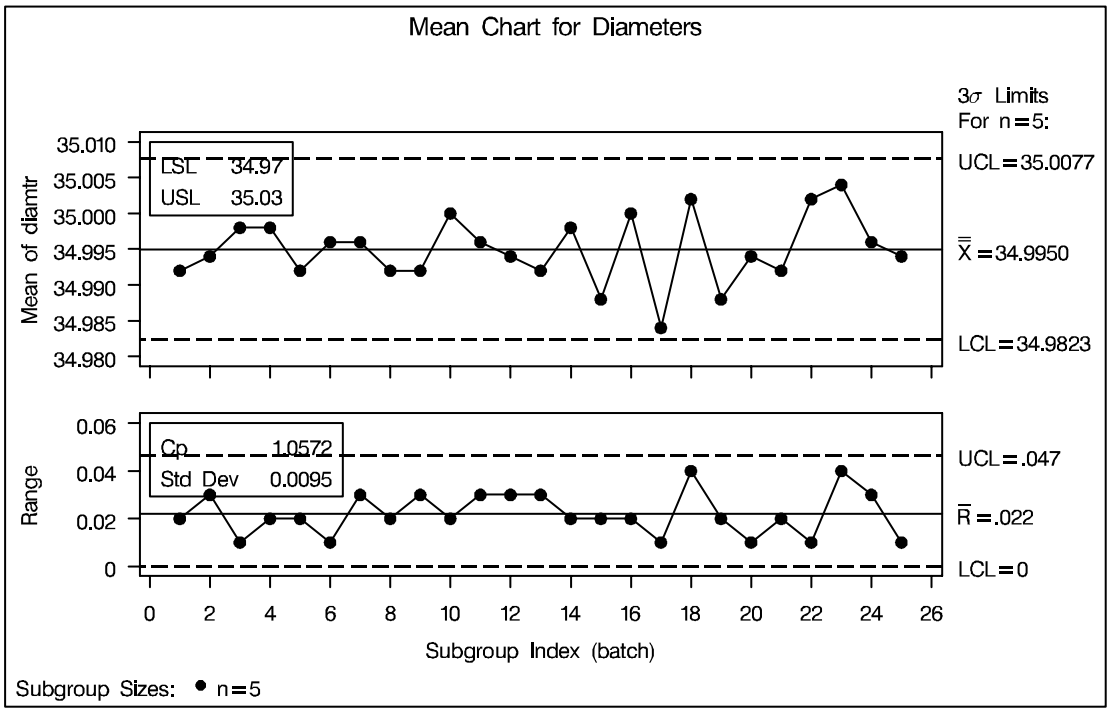


Figure 52.2. Formatting Values and Customizing Labels in an Inset

Adding a Header and Positioning the Inset

In the previous examples, the insets are displayed in the upper left corners of the plots, the default position for insets added to control charts. You can control the inset position with the POSITION= option. In addition, you can display a header at the top of the inset with the HEADER= option. The following statements create a data set to be used with the INSET DATA= keyword and the chart shown in Figure 52.3:

```
data location;
   length _LABEL_ $ 10 _VALUE_ $ 12;
   input _LABEL_ _VALUE_ &;
   datalines;
Plant      Santa Clara
Line       1
Shift      2
;

title 'Mean Chart for Diameters';
proc shewhart data=wafers;
   xchart diamtr*batch /
      lsl = 34.97
      usl = 35.03;
   inset data= location lsl='LSL' usl='USL' cp (6.4) stddev (6.4) /
      position = rm
      cshadow  = black
      header   = 'Summary Statistics';
run;
```

The header (in this case, *Summary Statistics*) can be up to 40 characters. Note that a relatively long list of inset statistics is requested. Consequently, POSITION=RM is specified to position the inset in the right margin. For more information about positioning, see "Details" on page 1847. The CSHADOW= option is used to display a drop shadow on this inset. The *options*, such as HEADER=, POSITION= and CSHADOW= are specified after the slash (/) in the INSET statement. For more details on INSET statement options, see "Dictionary of Options" on page 1845.

Note that the contents of the data set LOCATION appear before other statistics in the inset. The position of the DATA= keyword in the keyword list determines the position of the data set's contents in the inset.

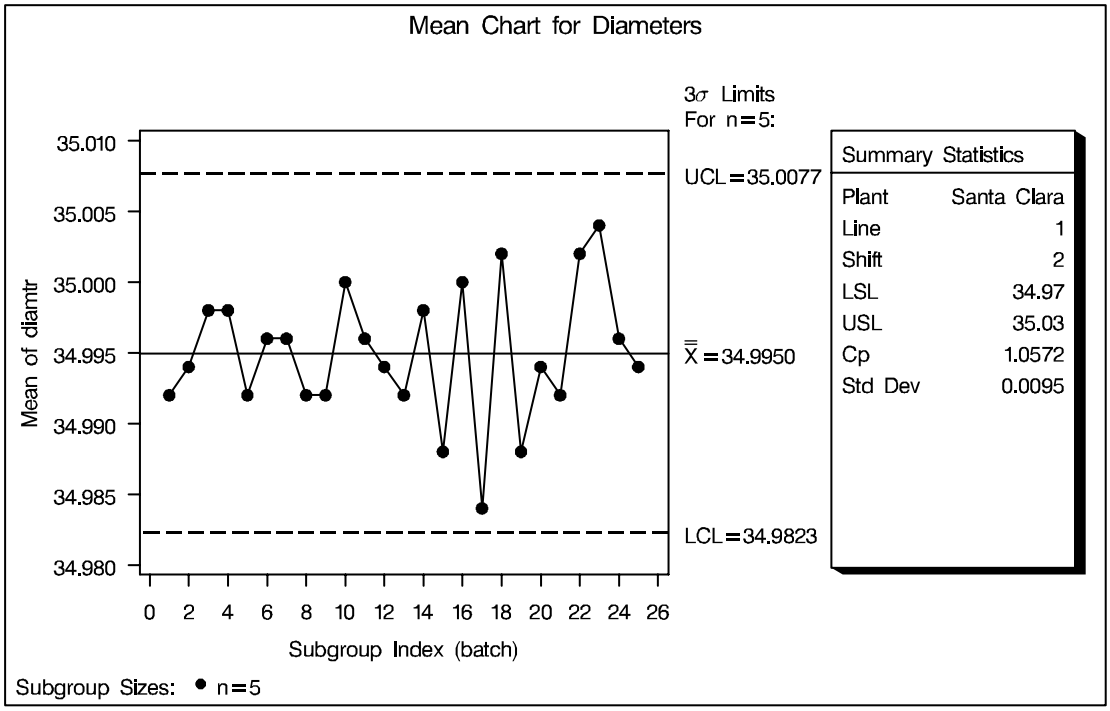


Figure 52.3. Adding a Header and Repositioning the Inset

Syntax

The syntax for the INSET and INSET2 statements is as follows:

INSET *keyword-list* < **/** *options* >;
INSET2 *keyword-list* < **/** *options* >;

You can use any number of INSET and INSET2 statements in the SHEWHART procedure. Each INSET or INSET2 statement produces a separate inset and must follow one of the chart statements. The inset appears on every panel (page) produced by the last chart statement preceding it. The statistics are displayed in the order in which they are specified. The following statements produce a boxplot with two insets and an $\bar{X}$ and R chart with one inset in the $\bar{X}$ chart and one in the R chart.

```
proc shewhart data=wafers;
   boxchart diamtr * batch / lsl=34.9 target=35 usl=35.1;
      inset lsl target usl;
      inset cp cpk cpm;
   xrchart diamtr*batch;
      inset nmin nmax nout;
      inset2 nlow2 nhigh2;
run;
```

The statistics displayed in an inset are computed for a specific process variable using observations for the current BY group. For example, in the following statements, there are two process variables (WEIGHT and DIAMETER) and a BY variable (LOCATION). If there are three different locations (levels of LOCATION), then a total of six $\bar{X}$ charts are produced. The statistics in each inset are computed for a particular variable and location. The labels in the inset are the same for each $\bar{X}$ chart.

```
proc shewhart data=axles;
   by location;
   xchart (weight diameter) * batch / tests=1 to 8;
   inset ntests 1 to 8;
run;
```

The components of the INSET and INSET2 statements are described as follows.

keyword-list
can include any of the *keywords* listed in "Summary of INSET Keywords" on page 1842. Some *keywords*, such as NTESTS and DATA=, require operands specified immediately after the *keyword*. Also, some inset statistics are available only if you request chart statements and options for which those statistics are calculated. For example,

- the NHIGH2, NLOW2, NTESTS2, LCL2 and UCL2 keywords are available only when a secondary chart is produced with the IRCHART, MRCHART, XRCHART or XSCHART statements.

- the NTESTS *keyword* requires the TESTS= option;
- the NTESTS2 *keyword* requires the TESTS2= option;
- the capability index *keywords* such as CPK all require one or more of the LSL=, USL= and TARGET= options.

By default, inset statistics are identified with appropriate labels, and numeric values are printed using appropriate formats. However, you can provide customized labels and formats. You provide the customized label by specifying the *keyword* for that statistic followed by an equal sign (=) and the label in quotes. Labels can have up to 24 characters. You provide the numeric format in parentheses after the *keyword*. Note that if you specify both a label and a format for a statistic, the label must appear before the format. For an example, see "Formatting Values and Customizing Labels" on page 1838.

options

appear after the slash (/) and control the appearance of the inset. For example, the following INSET statement uses two appearance *options* (POSITION= and CTEXT=):

```
inset n nmin nmax / position=ne ctext=yellow;
```

The POSITION= option determines the location of the inset, and the CTEXT= option specifies the color of the text of the inset.

See "Summary of Options" on page 1844 for a list of all available *options*, and "Dictionary of Options" on page 1845 for detailed descriptions. Note the difference between *keywords* and *options*; *keywords* specify the information to be displayed in an inset, whereas *options* control the appearance of the inset.

Summary of INSET Keywords

All keywords available with the SHEWHART procedure's INSET and INSET2 statements request a single statistic in an inset, except for the NTESTS, NTESTS2 and DATA= keywords. The NTESTS and NTESTS2 keywords each require a list of indexes specifying the tests for special causes whose counts of positive results are to be displayed:

```
inset ntests 1 2 3 4;
inset ntests2 1 to 4;
```

For each of the requested tests, the number of positive results for the test is displayed in the inset. So if tests 1 through 4 are requested the results occupy four lines in the inset.

The DATA= keyword specifies a SAS data set containing (label, value) pairs to be displayed in an inset. The data set must contain the variables _LABEL_ and _VALUE_. _LABEL_ is a character variable whose values provide labels for inset entries. _VALUE_ can be character or numeric, and provides values displayed in the

inset. The label and value from each observation in the DATA= data set occupy one line in the inset. Figure 52.3 shows an inset containing entries from a DATA= data set.

Table 52.1. Summary Statistics

DATA=	(label, value) pairs from *SAS-data-set*
LCL	primary chart lower control limit
MEAN	estimated or specified process mean
N	nominal subgroup size
NMIN	minimum subgroup size
NMAX	maximum subgroup size
NOUT	number of subgroups outside control limits on primary chart
NLOW	number of subgroups below lower control limit on primary chart
NHIGH	number of subgroups above upper control limit on primary chart
NTESTS	number of positive results of tests for special causes on primary chart
STDDEV	estimated or specified process standard deviation
UCL	primary chart lower control limit

Table 52.2. Secondary Chart Summary Statistics

LCL2	secondary chart lower control limit
MEAN2	mean of subgroup ranges or standard deviations
NOUT2	number of subgroups outside control limits on secondary chart
NLOW2	number of subgroups below lower control limit on secondary chart
NHIGH2	number of subgroups above upper control limit on secondary chart
NTESTS2	number of positive results of tests for special causes on secondary chart
UCL2	secondary chart upper control limit

Table 52.3. Specification Limits

LSL	lower specification limit
USL	upper specification limit
TARGET	target value

Table 52.4. Capability Indices and Confidence Limits

CIALPHA	α value for computing capability index confidence limits
CP	capability index C_p
CPLCL	lower confidence limit for C_p
CPUCL	upper confidence limit for C_p
CPK	capability index C_{pk}
CPKLCL	lower confidence limit for C_{pk}
CPKUCL	upper confidence limit for C_{pk}
CPL	capability index *CPL*
CPLLCL	lower confidence limit for *CPL*
CPLUCL	upper confidence limit for *CPL*
CPM	capability index C_{pm}
CPMLCL	lower confidence limit for C_{pm}
CPMUCL	upper confidence interval for C_{pm}
CPU	capability index *CPU*
CPULCL	lower confidence limit for *CPU*
CPUUCL	upper confidence limit for *CPU*

Summary of Options

The following table lists the INSET and INSET2 statement options. For complete descriptions, see "Dictionary of Options," which follows this section.

Table 52.5. INSET Options

CFILL=*color* | BLANK	specifies color of inset background
CFILLH=*color*	specifies color of header background
CFRAME=*color*	specifies color of frame
CHEADER=*color*	specifies color of header text
CSHADOW=*color*	specifies color of drop shadow
CTEXT=*color*	specifies color of inset text
DATA	specifies data units for POSITION=(x, y) coordinates
FONT=*font*	specifies font of text
FORMAT=*format*	specifies format of values in inset
HEADER=*'quoted string'*	specifies header text
HEIGHT=*value*	specifies height of inset text
NOFRAME	suppresses frame around inset
POSITION=*position*	specifies position of inset
REFPOINT=BR|BL|TR|TL	specifies reference point of inset positioned with POSITION=(x, y) coordinates

Dictionary of Options

The following entries provide detailed descriptions of options for the INSET and INSET2 statements. Terms used in this section are illustrated in Figure 52.4.

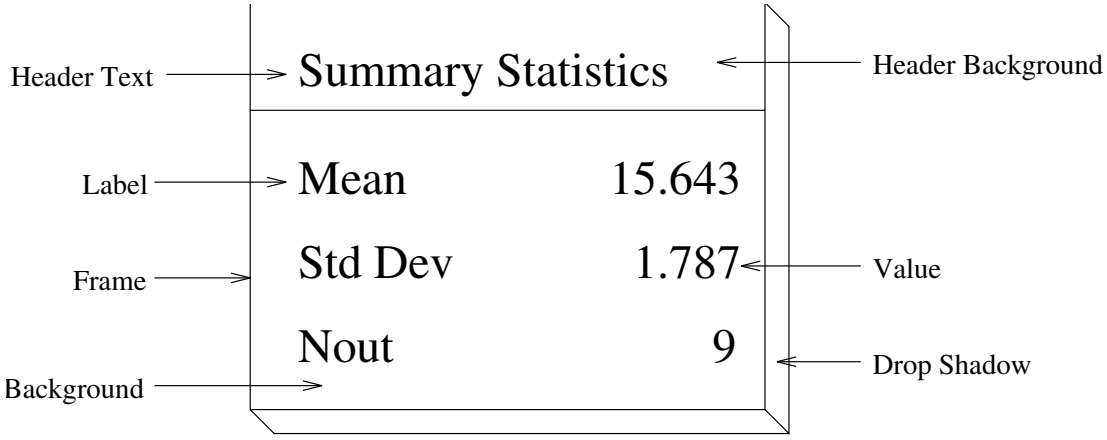


Figure 52.4. The Inset

CFILL=*color* **| BLANK**
specifies the color of the background (including the header background if you do not specify the CFILLH= option).

If you do not specify the CFILL= option, then by default, the background is empty. This means that items that overlap the inset (such as subgroup data points or control limits) show through the inset. If you specify any value for the CFILL= option, then overlapping items no longer show through the inset. Specify CFILL=BLANK to leave the background uncolored and also to prevent items from showing through the inset.

CFILLH=*color*
specifies the color of the header background. By default, if you do not specify a CFILLH= color, the CFILL= color is used.

CFRAME=*color*
specifies the color of the frame. By default, the frame is the same color as the axis of the plot.

CHEADER=*color*
specifies the color of the header text. By default, if you do not specify a CHEADER= color, the CTEXT= color is used.

CSHADOW=*color*
CS=*color*
specifies the color of the drop shadow. See Figure 52.3 on page 1840 for an example. By default, if you do not specify the CSHADOW= option, a drop shadow is not displayed.

CTEXT=*color*
CT=*color*
specifies the color of the text. By default, the inset text color is the same as the other text on the plot.

DATA
specifies that data coordinates are to be used in positioning the inset with the POSITION= option. The DATA option is available only when you specify POSITION= (x, y), and it must be placed immediately after the coordinates (x, y). For details, see the entry for the POSITION= option or "Positioning the Inset Using Coordinates" on page 1848. See Figure 52.7 on page 1849 for an example.

FONT=*font*
specifies the font of the text. By default, the font is SIMPLEX if the inset is located in the interior of the plot, and the font is the same as the other text displayed on the plot if the inset is located in the exterior of the plot.

FORMAT=*format*
specifies a format for all the values displayed in an inset. If you specify a format for a particular statistic, then this format overrides the format you specified with the FORMAT= option.

HEADER= *'string'*
specifies the header text. The *string* cannot exceed 40 characters. If you do not specify the HEADER= option, no header line appears in the inset.

HEIGHT=*value*
specifies the height of the text.

NOFRAME
suppresses the frame drawn around the text.

POSITION=*position*
POS=*position*
determines the position of the inset. The *position* can be a compass point keyword, a margin keyword, or a pair of coordinates (x, y). You can specify coordinates in axis percent units or axis data units. For more information, see "Details" on page 1847. By default, POSITION=NW, which positions the inset in the upper left (northwest) corner of the display.

REFPOINT=BR | BL | TR | TL
RP=BR | BL | TR | TL
specifies the reference point for an inset that is positioned by a pair of coordinates with the POSITION= option. Use the REFPOINT= option with POSITION= coordinates. The REFPOINT= option specifies which corner of the inset frame you want positioned at coordinates (x, y). The keywords BL, BR, TL, and TR represent bottom left, bottom right, top left, and top right, respectively. See Figure 52.8 on page 1850 for an example. The default is REFPOINT=BL.

If you specify the position of the inset as a compass point or margin keyword, the REFPOINT= option is ignored. For more information, see "Positioning the Inset Using Coordinates" on page 1848.

Details

This section provides details on three different methods of positioning the inset using the POSITION= option. With the POSITION= option, you can specify

- compass points
- keywords for margin positions
- coordinates in data units or percent axis units

Positioning the Inset Using Compass Points

You can specify the eight compass points N, NE, E, SE, S, SW, W, and NW as keywords for the POSITION= option. The following statements create the display in Figure 52.5, which demonstrates all eight compass positions. The default is NW.

```
title 'Mean Chart for Diameters';
proc shewhart data=wafers;
   xchart diamtr*batch / tests= 1 to 8;
   inset ntests 1 / height=3 cfill=blank header='NW' pos=nw;
   inset ntests 2 / height=3 cfill=blank header='N ' pos=n ;
   inset ntests 3 / height=3 cfill=blank header='NE' pos=ne;
   inset ntests 4 / height=3 cfill=blank header='E ' pos=e ;
   inset ntests 5 / height=3 cfill=blank header='SE' pos=se;
   inset ntests 6 / height=3 cfill=blank header='S ' pos=s ;
   inset ntests 7 / height=3 cfill=blank header='SW' pos=sw;
   inset ntests 8 / height=3 cfill=blank header='W ' pos=w ;
run;
```

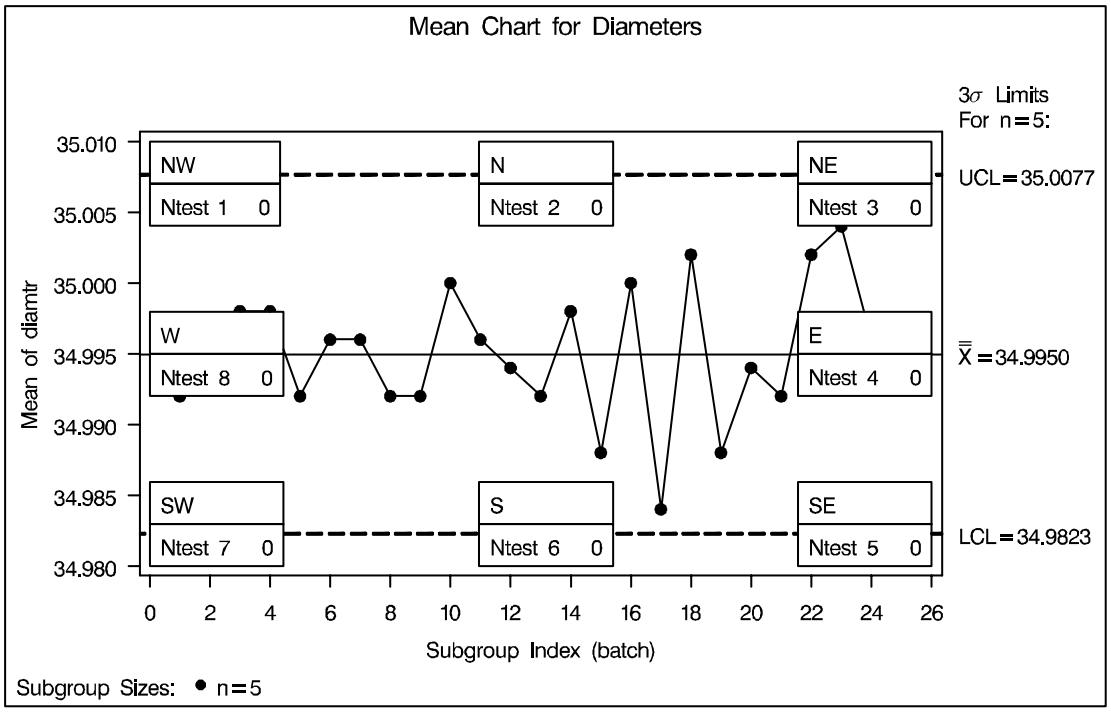


Figure 52.5. Insets Positioned Using Compass Points

Positioning the Inset in the Margins

Using the INSET statement you can also position an inset in one of the four margins surrounding the plot area using the margin keywords LM, RM, TM, or BM, as illustrated in Figure 52.6. The INSET2 statement cannot be used to produce an inset in a margin.

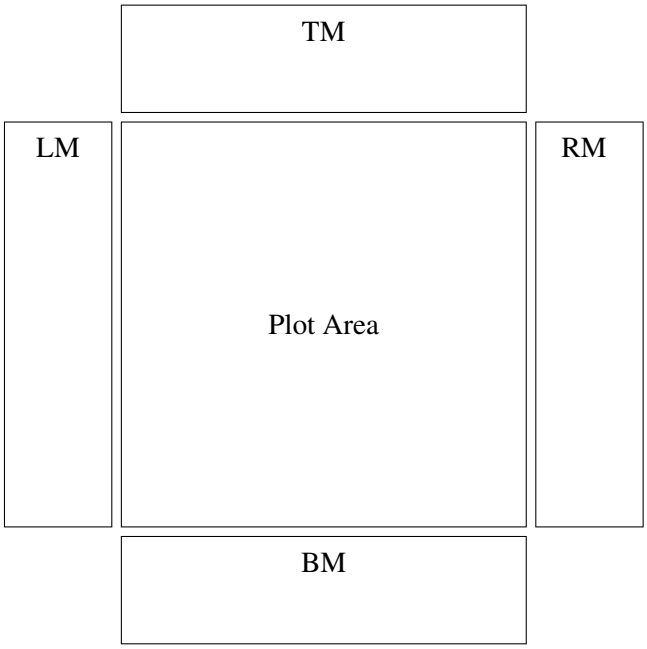


Figure 52.6. Positioning Insets in the Margins

For an example of an inset placed in the right margin, see Figure 52.3 on page 1840. Margin positions are recommended if a large number of statistics are listed in the INSET statement. If you attempt to display a lengthy inset in the interior of the plot, it is likely that the inset will collide with the data display.

Positioning the Inset Using Coordinates

You can also specify the position of the inset with coordinates: POSITION= (x, y). The coordinates can be given in axis percent units (the default) or in axis data units.

Data Unit Coordinates

If you specify the DATA option immediately following the coordinates, the inset is positioned using axis data units. For example, the following statements place the bottom left corner of the inset at 6 on the horizontal axis and 34.985 on the vertical axis:

```
title 'Mean Chart for Diameters';
proc shewhart data=wafers;
   xchart diamtr*batch;
   inset n /
      header   = 'Position=(6,34.985)'
      position = (6,34.985) data;
run;
```

The control chart is displayed in Figure 52.7. By default, the specified coordinates determine the position of the bottom left corner of the inset. You can change this reference point with the REFPOINT= option, as in the next example.

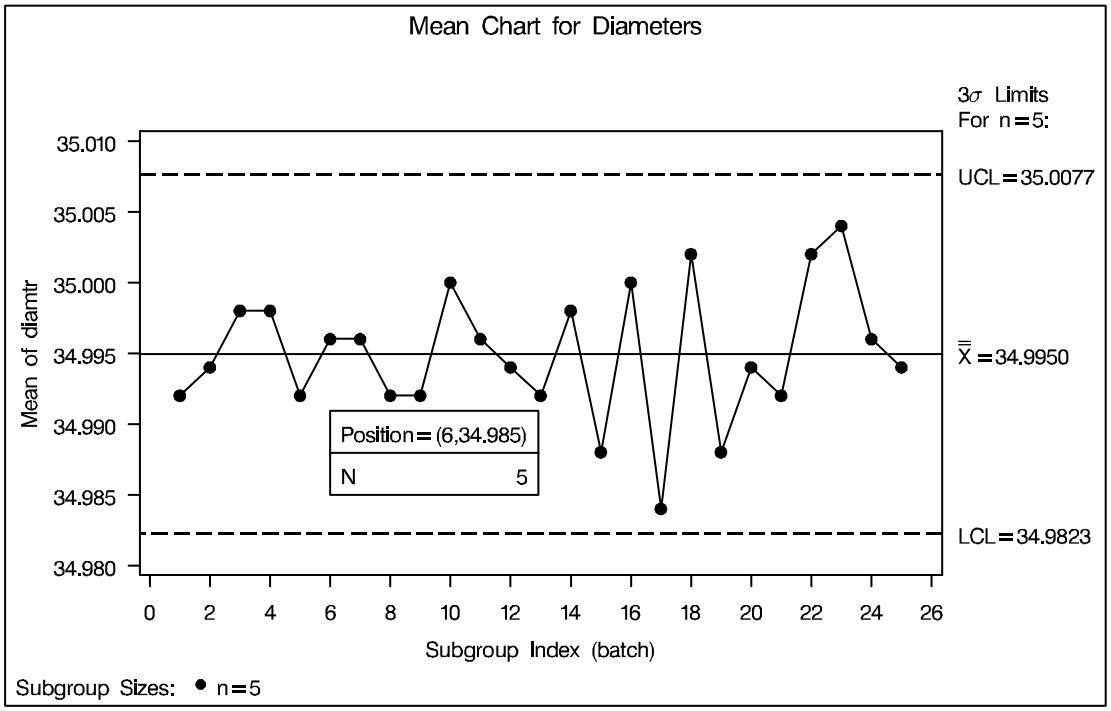


Figure 52.7. Inset Positioned Using Data Unit Coordinates

Axis Percent Unit Coordinates

If you do not use the DATA option, the inset is positioned using axis percent units. The coordinates of the bottom left corner of the display are $(0, 0)$, while the upper right corner is $(100, 100)$. For example, the following statements create a $\bar{X}$ chart with two insets, both positioned using coordinates in axis percent units:

```
title 'Mean Chart for Diameters';
proc shewhart data=wafers;
   xchart diamtr*batch;
   inset nmin / position = (5,25)
                header   = 'Position=(5,25)'
                height   = 3
                cfill    = blank
                refpoint = tl;
   inset nmax / position = (95,95)
                header   = 'Position=(95,95)'
                height   = 3
                cfill    = blank
                refpoint = tr;
run;
```

The display is shown in Figure 52.8. Notice that the REFPOINT= option is used to determine which corner of the inset is to be placed at the coordinates specified with the POSITION= option. The first inset has REFPOINT=TL, so the top left corner of the inset is positioned 5% of the way across the horizontal axis and 25% of the way up the vertical axis. The second inset has REFPOINT=TR, so the top right corner of the inset is positioned 95% of the way across the horizontal axis and 95% of the way up the vertical axis. Note also that coordinates in axis percent units must be *between* 0 and 100.

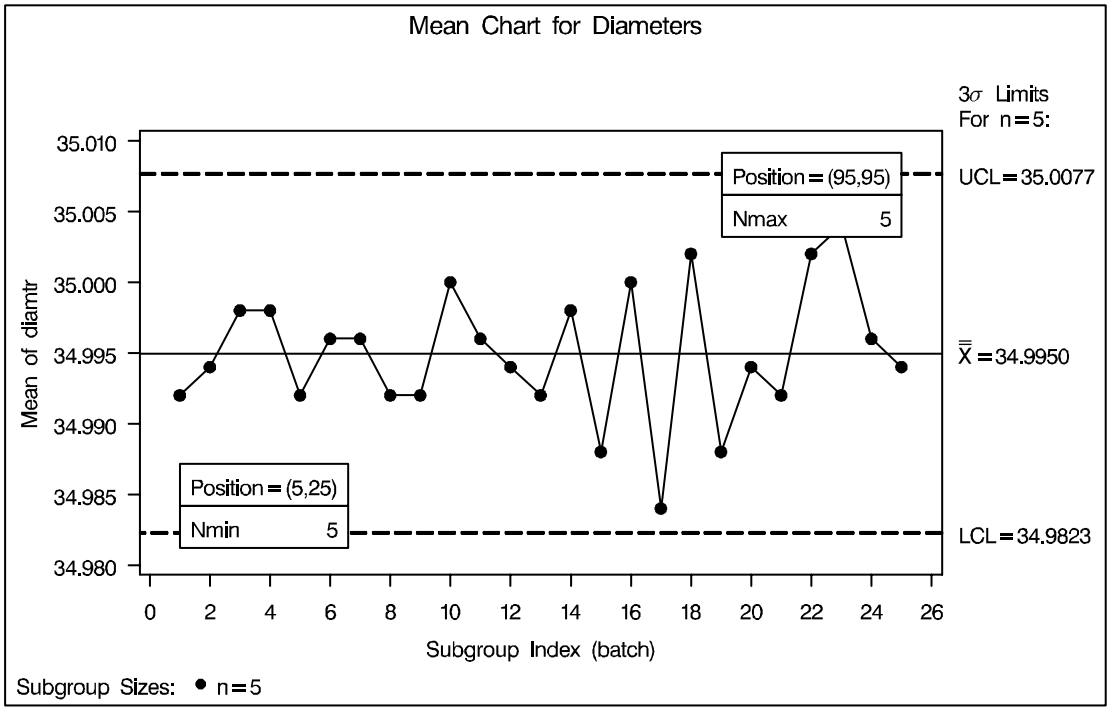


Figure 52.8. Inset Positioned Using Axis Percent Unit Coordinates

Chapter 53
Dictionary of Options

Chapter Contents

DICTIONARY OF OPTIONS 1853

Chapter 53
Dictionary of Options

Dictionary of Options

This chapter provides detailed descriptions of options that you can specify in the following chart statements:

- BOXCHART
- CCHART
- IRCHART
- MCHART
- MRCHART
- NPCHART
- PCHART
- RCHART
- SCHART
- UCHART
- XCHART
- XRCHART
- XSCHART

Options are specified after the slash (/) in a chart statement. For example, to request tests for special causes with an $\bar{X}$ and R chart, you can use the TESTS= option as follows:

```
proc shewhart data=measures;
   xrchart length*sample / tests=1 to 4 ;
run;
```

The options described in this chapter are listed alphabetically. For tables of options organized by function, see the "Summary of Options" sections in the chapters for the various chart statements.

Unless indicated otherwise, the options listed here are available with every chart statement. The marginal notes *Graphics* and *Line Printer* identify options that apply only to charts displayed with graphics devices and line printers, respectively. For statements that create two charts, the term *primary chart* refers to the upper chart (for instance, the $\bar{X}$ chart created with the XRCHART statement), and the term *secondary chart* refers to the lower chart (for instance, the R chart created with the XRCHART statement). The term *primary chart* also refers to the single chart created by some statements (for instance, the p chart created with the PCHART statement).

ALLLABEL=VALUE
ALLLABEL=(*variable***)**
labels every point on the primary chart with the VALUE plotted for that subgroup or with the value of a *variable* in the input data set.

The *variable* provided in the input data set can be numeric or character. If the *variable* is a character variable, its length cannot exceed 16. For each subgroup of observations, the formatted value of the *variable* in the observations is used to label the point representing the subgroup. If you are reading a DATA= data set with multiple observations per subgroup, the values of the *variable* should be identical for observations within a subgroup. You should use this option with care to avoid cluttering the chart. By default, points are not labeled. Related options are CFRAMELAB=, OUTLABEL=, LABELFONT=, LABELHEIGHT=, and TESTLABEL=, but note that the OUTLABEL= option cannot be specified with the ALLLABEL= option.

ALLLABEL2=VALUE
ALLLABEL2=(*variable***)**
labels every point on an R, s, or trend chart with the VALUE plotted for that subgroup or with the value of a *variable* in the input data set.

The *variable* provided in the input data set can be numeric or character. If the *variable* is a character variable, its length cannot exceed 16. For each subgroup of observations, the formatted value of the *variable* in the observations is used to label the point representing the subgroup. If you are reading a DATA= data set with multiple observations per subgroup, the values of the *variable* should be identical for observations within a subgroup. You should use this option with care to avoid cluttering the chart. By default, points are not labeled. Related options are CFRAMELAB=, OUTLABEL2=, LABELFONT=, LABELHEIGHT=, and TESTLABEL2=, but note that the OUTLABEL2= option cannot be specified with the ALLLABEL2= option. The option is available in the IRCHART, MRCHART, RCHART, SCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option.

ALLN
plots summary statistics for all subgroups, regardless of whether the subgroup sample size equals the nominal control limit sample size n specified by the LIMITN= option or the variable _LIMITN_ in the LIMITS= data set. Use the ALLN option in conjunction with the LIMITN= option or the variable _LIMITN_.

The ALLN option is useful in applications where almost all of the subgroups have a common sample size n, and you want to display fixed (rather than varying) control limits corresponding to the nominal sample size n. The disadvantage of using the ALLN option with widely differing subgroup sample sizes is that the interpretation of the control limits is meaningful only for those subgroups whose sample size is equal to n. To request special symbol markers indicating that not all the sample sizes are equal to n, use the NMARKERS option in conjunction with the ALLN option.

The ALLN option is not available in the IRCHART statement.

ALPHA=*value*

requests *probability limits*. If you specify ALPHA=α, the control limits are computed so that the probability is α that a subgroup summary statistic exceeds its control limits. This assumes that the process is in statistical control and that the data follow a certain theoretical distribution, which depends on the chart statement. The Poisson distribution is assumed for the CCHART and UCHART statements, and the binomial distribution is assumed for the NPCHART and PCHART statements. The normal distribution is assumed for all other chart statements. For the equations used to compute probability limits, see the "Details" section in the chapter for the chart statement that you are using.

The value of α can range between 0 and 1 for most statements. However, for the MCHART statement, the MRCHART statement, and the BOXCHART statement with the CONTROLSTAT=MEDIAN option, the value of α must be one of the following: 0.001, 0.002, 0.01, 0.02, 0.025, 0.04, 0.05, 0.10, or 0.20.

Note the following:

- As an alternative to specifying ALPHA=α, you can read α from the variable _ALPHA_ in a LIMITS= data set by specifying the READALPHA option. See "Input Data Sets" in the chapter for the chart statement in which you are interested.
- As an alternative to specifying ALPHA=α (or reading the variable _ALPHA_ from a LIMITS= data set), you can request "$k\sigma$ control limits" by specifying SIGMAS=k (or reading the variable _SIGMAS_ from a LIMITS= data set).

If you specify neither the ALPHA= option nor the SIGMAS= option, the procedure computes 3σ control limits by default.

ANNOTATE=*SAS-data-set*
ANNO=*SAS-data-set*

specifies an ANNOTATE= type data set, as described in *SAS/GRAPH Software: Reference*, that enhances a primary chart. The ANNOTATE= data set specified in a chart statement enhances all charts created by that particular statement. You can also specify an ANNOTATE= data set in the PROC SHEWHART statement to enhance all primary charts created by the procedure. *Graphics*

ANNOTATE2=*SAS-data-set*
ANNO2=*SAS-data-set*

specifies an ANNOTATE= type data set, as described in *SAS/GRAPH Software: Reference*, that enhances a secondary chart. The ANNOTATE2= data set specified in a chart statement enhances all charts created by that particular statement. You can also specify an ANNOTATE2= data set in the PROC SHEWHART statement to enhance all secondary charts created by the procedure. *Graphics*

This option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option.

BILEVEL

arranges the Shewhart chart in two levels (rather than the default of one level) so that twice as much data can be displayed on a page or screen. The second level is a continuation of the first level, and this arrangement is continued on subsequent pages until all the subgroups are displayed. You use the NPANELPOS= option to control the number of subgroup positions in each level. If you specify the BILEVEL option in a chart statement that produces primary and secondary charts, you must also specify the SEPARATE option.

BLOCKLABELPOS=ABOVE | LEFT | RIGHT

specifies the position of a block-variable label in the block legend. As shown in Figure 53.1, the keyword ABOVE places the label immediately above the legend, LEFT places the label to the left of the legend, and RIGHT places the label to the right of the legend. Use the keywords LEFT and RIGHT with labels that are short enough to fit in the margins on each side of the chart; otherwise, they will be truncated. Use the keyword RIGHT only when the legend is below the control chart (BLOCKPOS=3 or BLOCKPOS=4). The default keyword is ABOVE. Related options are BLOCKLABTYPE=, BLOCKREP, BLOCKPOS=, CBLOCKVAR=, and CBLOCKLAB=.

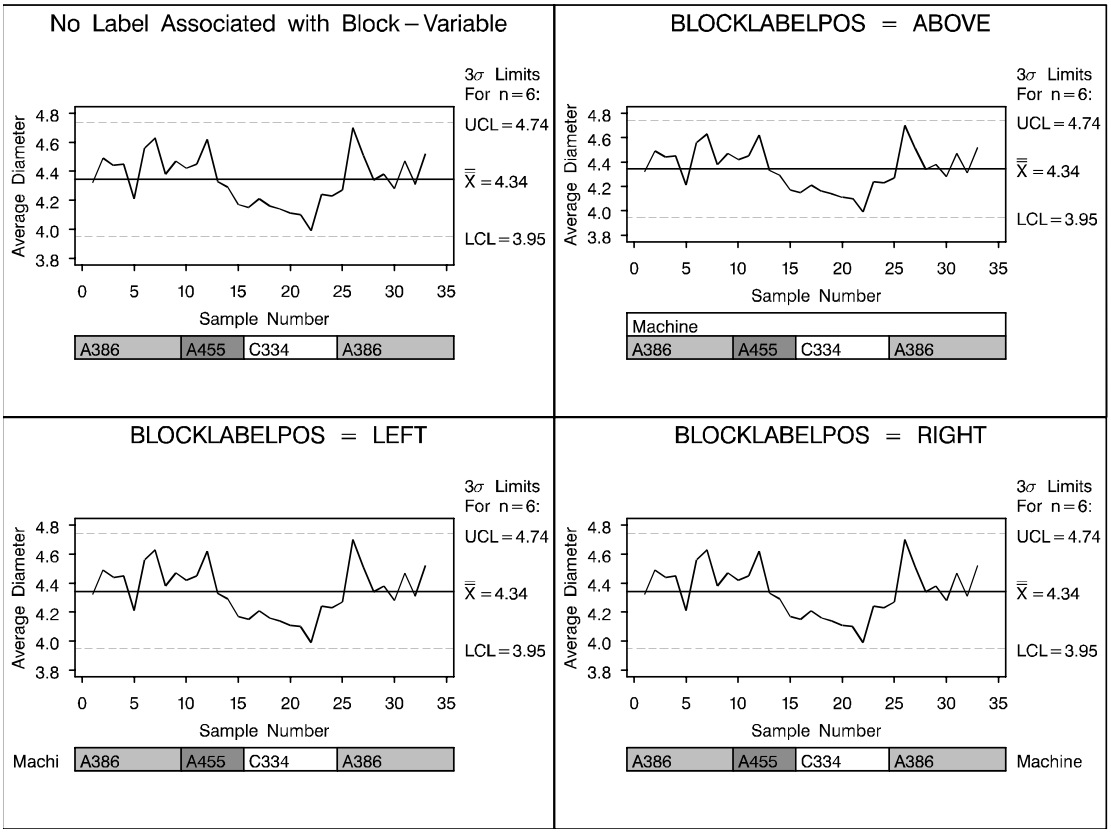


Figure 53.1. Positions for *block-variable* Labels

BLOCKLABTYPE=SCALED | TRUNCATED
BLOCKLABTYPE=*height*

Graphics

specifies how lengthy block variable values are to be treated when there is insufficient space to display them in the block legend. If you specify the BLOCKLABTYPE=SCALED option, the values are uniformly reduced in height so that they fit. If you specify the BLOCKLABTYPE=TRUNCATED option, lengthy values are truncated on the right until they fit. You can also specify a text *height* in vertical percent screen units for the values. By default, lengthy values are not displayed. Related options are BLOCKLABELPOS=, BLOCKREP, BLOCKPOS=, CBLOCKVAR=, and CBLOCKLAB=.

BLOCKPOS=*n*

specifies the vertical position of the legend for the values of the *block-variables* (see "Displaying Stratification in Blocks of Observations" on page 1932). Values of n and the corresponding positions are as follows. By default, BLOCKPOS=1.

n	Legend Position
1	Top of chart, offset from axis frame
2	Top of chart, immediately above axis frame
3	Bottom of chart, immediately above horizontal axis
4	Bottom of chart, below horizontal axis label

Figure 53.2 illustrates the various positions that can be specified.

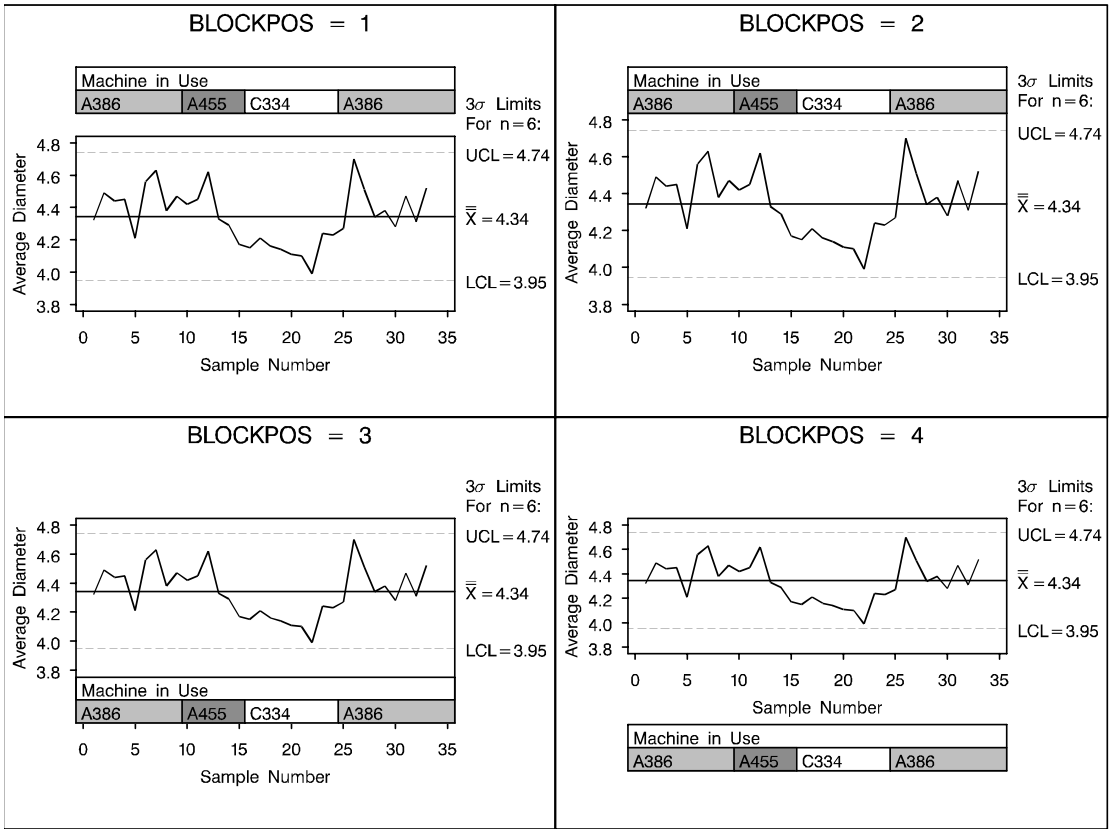


Figure 53.2. Positions for *block-variable* Legends

Related options are BLOCKLABELPOS=, BLOCKLABTYPE=, BLOCKREP, CBLOCKVAR=, and CBLOCKLAB=.

BLOCKREP

specifies that block variable values for all subgroups are to be displayed. By default, only the first block variable value in any block is displayed, and repeated block variable values are not displayed. Related options are BLOCKLABELPOS=, BLOCKLABTYPE=, BLOCKPOS=, CBLOCKVAR=, and CBLOCKLAB=. For more information on block variables, see "Displaying Stratification in Blocks of Observations" on page 1932.

BOXCONNECT
BOXCONNECT=MEAN | MEDIAN | MAX | MIN | Q1 | Q3

specifies that the points representing subgroup means, medians, maximum values, minimum values, first quartiles or third quartiles in box-and-whisker plots created with the BOXCHART statement are to be connected. If BOXCONNECT is specified without a keyword identifying the points to be connected, subgroup means are connected. By default, no points are connected. The BOXCONNECT option is available only in the BOXCHART statement.

BOXSTYLE=*keyword*

specifies the style of the box-and-whisker plots that are displayed for subgroup samples by the BOXCHART statement.

The keywords SKELETAL, SCHEMATIC, SCHEMATICID, and SCHEMATICIDFAR are useful for creating conventional box-and-whisker displays. The keywords POINTS, POINTSJOIN, POINTSBOX, POINTSID, and POINTSJOINID are used to generalize the BOXSTYLE= option and, in particular, to facilitate the creation of so-called "multi-vari" charts, as illustrated in Output 39.7.2 and Output 39.7.3. The keyword POINTSSCHEMATIC combines the POINT and SCHEMATIC boxstyles.

If you specify BOXSTYLE=SKELETAL, the whiskers are drawn from the edges of the box to the extreme values of the subgroup sample. This plot is sometimes referred to as a *skeletal box-and-whisker plot*. By default, the whiskers are drawn without serifs, but you can add serifs with the SERIFS option. Figure 53.3 illustrates the elements of a typical skeletal boxplot.

If you specify BOXSTYLE=SCHEMATIC, a whisker is drawn from the upper edge of the box to the largest value within the upper fence and from the lower edge of the box to the smallest value within the lower fence. Figure 53.4 illustrates a typical schematic boxplot and the locations of the fences (which are not displayed in actual output). Serifs are added to the whiskers by default. Observations outside the fences are identified with a special symbol; you can specify the shape and color for this symbol with the IDSYMBOL= and IDCOLOR= options. The default symbol is a square. This type of plot corresponds to the *schematic box-and-whisker plot* described in Chapter 2 of Tukey (1977).

If you specify BOXSTYLE=SCHEMATICID, a schematic box-and-whisker plot is displayed in which the value of the first variable listed in the ID statement is used to label the symbol marking each observation outside the upper and lower fences.

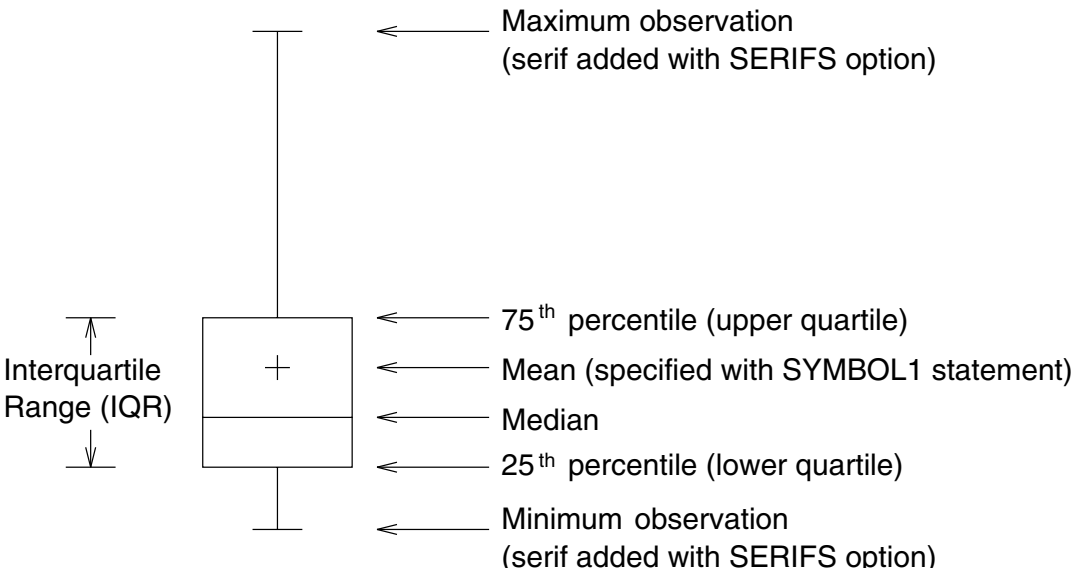


Figure 53.3. BOXSTYLE= SKELETAL

If you specify BOXSTYLE=SCHEMATICIDFAR, a schematic box-and-whisker plot is displayed in which the value of the first variable listed in the ID statement is used to label the symbol marking each observation outside the *lower* and *upper far fences*. The lower and upper far fences are located $3\times$IQR below the 25^{th} percentile and above the 75^{th} percentile, respectively. Observations between the fences and the far fences are identified with a symbol but are not labeled with the ID variable.

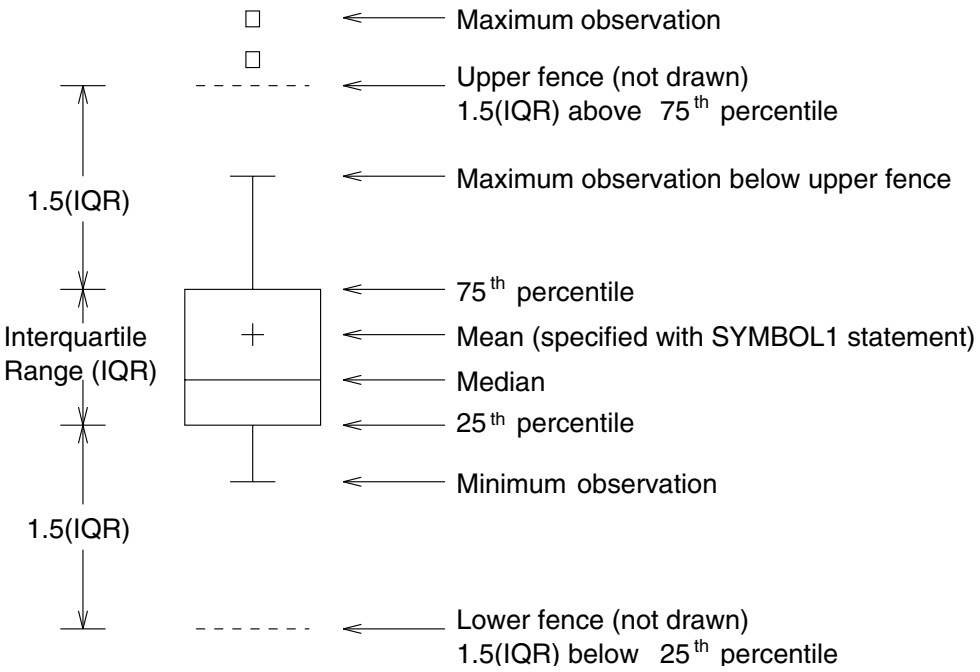


Figure 53.4. BOXSTYLE= SCHEMATIC

Note: To make side-by-side box charts (as opposed to a control chart with subgroup box plots), you should use the BOXCHART statement with the NOLIMITS option in addition to the BOXSTYLE= option.

Graphics

If you specify BOXSTYLE=POINTS, all the values in the subgroup sample are plotted as points, and neither a box nor whiskers are drawn. By default, a square plotting symbol is used for the values. You can specify a symbol with the IDSYMBOL= option. You can specify the color of the symbols with the IDCOLOR= option (the default color is the color specified with the CBOXES= option or the second color in the device color list).

Graphics

If you specify BOXSTYLE=POINTSJOIN, all the values in the subgroup sample are plotted as points joined with a vertical line. Neither a box nor whiskers are drawn. See Output 39.7.2 on page 1300 for an illustration. By default, a square plotting symbol is used for the values. You can specify a symbol with the IDSYMBOL= option, and you can specify the color of the symbol with the IDCOLOR= option. You can specify the color of the vertical line with the CBOXES= option.

Graphics

If you specify BOXSTYLE=POINTSBOX, all the values in the subgroup sample are plotted as points enclosed in a box. By default, a square plotting symbol is used for the values. You can specify a symbol with the IDSYMBOL= option, and you can specify the color of the symbol with the IDCOLOR= option. You can specify the color of the box with the CBOXES= option, the fill color of the box with the CBOXFILL= option, and the line type of the box with the LBOXES= option.

Graphics

If you specify BOXSTYLE=POINTSID, all the values in the subgroup sample are plotted using labels specified as the values of the first variable in the ID statement. See Output 39.7.3 on page 1302 for an illustration. It is recommended that you use single-character labels. You can specify a font for the labels with the IDFONT= option. You can specify the height of the labels with the IDHEIGHT= option. You can specify the color of the labels with the IDCTEXT= option.

Graphics

If you specify BOXSTYLE=POINTSJOINID, all the values in the subgroup sample are plotted using labels specified as the values of the first variable in the ID statement, and the values are joined by a vertical line. It is recommended that you use single-character labels. You can specify a font for the labels with the IDFONT= option. You can specify the height of the labels with the IDHEIGHT= option. You can specify the color of the labels with the IDCTEXT= option, and you can specify the color of the vertical line with the CBOXES= option.

Graphics

If you specify BOXSTYLE=POINTSSCHEMATIC, a schematic box chart is overlaid with points plotting all observations in the subgroups.

The BOXSTYLE= option is available only in the BOXCHART statement; see Example 39.2 on page 1287. The styles SCHEMATIC, SCHEMATICID, and SCHEMATICIDFAR are available only when the input data set is a DATA= data set. By default, BOXSTYLE= SKELETAL. Related options include BOXWIDTH=, BOXWIDTHSCALE=, IDCOLOR=, and IDSYMBOL=.

Note that the keywords POINTS, POINTSJOIN, POINTSBOX, POINTSID, and POINTSJOINID for the BOXSTYLE= option can be used in conjunction with the CPHASEBOX=, CPHASEBOXFILL=, CPHASEBOXCONNECT=, CPHASEMEANCONNECT=, and PHASEMEANSYMBOL= options to create "multi-vari" displays.

BOXWIDTH=*value*

specifies the width (in horizontal percent screen units) of box-and-whisker plots created with the BOXCHART statement. The default width is chosen so that the boxes are as wide as possible without colliding. You should use the BOXWIDTH= option in situations where the number of subgroups per panel is very small and you want to reduce the width. The BOXWIDTH= option is available only in the BOXCHART statement. *Graphics*

BOXWIDTHSCALE=*value*

specifies that the width of box-and-whisker plots created with the BOXCHART statement is to vary proportionately to a particular function of the subgroup sample size n. The function is determined by the *value* and is identified on the chart with a legend. *Graphics*

If you specify a positive *value*, the widths are proportional to n^{value}. In particular, if you specify BOXWIDTHSCALE=1, the widths are proportional to the sample size. If you specify BOXWIDTHSCALE=0.5, the widths are proportional to $\sqrt{n}$, as described by McGill and others (1978). If you specify BOXWIDTHSCALE=0, the widths are proportional to $\log(n)$. See Example 39.4 on page 1292 for an illustration of the BOXWIDTHSCALE= option.

By default, the box widths are constant. The BOXWIDTHSCALE= option is available only in the BOXCHART statement.

CAXIS=*color*
CAXES=*color*
CA=*color*

specifies the color for the axes and tick marks. This option overrides any COLOR= specifications in an AXIS statement. The default is the first color in the device color list. *Graphics*

CBLOCKLAB=*color* **|** **(***color-list***)**

specifies fill colors for the frames that enclose the *block-variable* labels in a block legend. By default, these areas are not filled. Colors in the CBLOCKLAB= list are matched with *block-variables* in the order in which they appear in the chart statement. Related options are BLOCKLABELPOS=, BLOCKLABTYPE=, BLOCKREP, BLOCKPOS=, and CBLOCKVAR=. *Graphics*

CBLOCKVAR=*variable* **|** **(***variable-list***)**

specifies variables whose values are colors for filling the background of the legend associated with *block-variables*. Each CBLOCKVAR= variable must be a character variable of no more than eight characters in the input data set (a DATA=, HISTORY=, or TABLE= data set). A list of CBLOCKVAR= variables must be enclosed in parentheses. *Graphics*

The procedure matches the CBLOCKVAR= variables with *block-variables* in the order specified. That is, each block legend will be filled with the color value of the CBLOCKVAR= variable of the first observation in each block. In general, values of the i^{th} CBLOCKVAR= variable are used to fill the block of the legend corresponding to the i^{th} *block-variable*. For examples of the CBLOCKVAR= option, see Figure 54.4 on page 1935 and Figure 54.5 on page 1936.

By default, fill colors are not used for the *block-variable* legend. The CBLOCKVAR= option is available only when *block-variables* are used in the chart statement.

CBOXES=*color*
CBOXES=(*variable***)**

Graphics

specifies the colors for the outlines of the box-and-whisker plots created with the BOXCHART statement. You can use one of the following approaches:

- You can specify CBOXES=*color* to provide a single outline color for all the box-and-whisker plots.
- You can specify CBOXES=(*variable*) to provide a distinct outline color for *each* box-and-whisker plot as the value of the *variable*. The *variable* must be a character variable of length 8 less in the input data set, and its values must be valid SAS/GRAPH color names. The outline color of the plot displayed for a particular subgroup is the value of the *variable* in the observations corresponding to this subgroup. Note that if there are multiple observations per subgroup in the input data set, the values of the *variable* should be identical for all the observations in a given subgroup.

The default *color* is the second color in the device color list. The CBOXES= option is available only in the BOXCHART statement.

CBOXFILL=*color*
CBOXFILL=(*variable***)**

Graphics

specifies the interior fill colors for the box-and-whisker plots created with the BOXCHART statement. You can use one of the following approaches:

- You can specify CBOXFILL=*color* to provide a single color for all of the box-and-whisker plots.
- You can specify CBOXFILL=(*variable*) to provide a distinct color for *each* box-and-whisker plot as the value of the *variable*. The *variable* must be a character variable of length 8 or less in the input data set, and its values must be valid SAS/GRAPH color names (or the value *EMPTY*, which you can use to suppress color filling). The interior color of the plot displayed for a particular subgroup is the value of the *variable* in the observations corresponding to this subgroup. Note that if there are multiple observations per subgroup in the input data set, the values of the *variable* should be identical for all the observations in a given subgroup.

By default, the interiors are not filled. The CBOXFILL= option is available only in the BOXCHART statement.

CCLIP=*color*

specifies a color for the plotting symbol that is specified with the CLIPSYMBOL= option to mark clipped points. The default color is the color specified in the COLOR= option in the SYMBOL1 statement. *Graphics*

CCONNECT=*color*

specifies the color for the line segments connecting points on the chart. The default color is the color specified in the COLOR= option in the SYMBOL1 statement. This option is not applicable in the BOXCHART statement unless you also specify the BOXCONNECT option. *Graphics*

CCOVERLAY=(*color-list***)**

specifies the colors for the line segments connecting points on primary chart overlays. Colors in the CCOVERLAY= list are matched with variables in the corresponding positions in the OVERLAY= list. By default, points are connected by line segments of the same color as the plotted points. You can specify the value NONE to suppress the line segments connecting points on an overlay. *Graphics*

CCOVERLAY2=(*color-list***)**

specifies the colors for the line segments connecting points on secondary chart overlays. Colors in the CCOVERLAY2= list are matched with variables in the corresponding positions in the OVERLAY2= list. By default, points are connected by line segments of the same color as the plotted points. You can specify the value NONE to suppress the line segments connecting points on an overlay. *Graphics*

CFRAME=*color*

CFRAME=(*color-list***)**

specifies the colors for filling the rectangle enclosed by the axes and the frame. By default, this area is not filled. The CFRAME= option cannot be used in conjunction with the NOFRAME option. *Graphics*

You can specify a single *color* to fill the entire area. Alternatively, if you are displaying phases (blocks) of data read with the READPHASES= option, you can specify a *color-list* with the CFRAME= option to fill the sub-rectangles of the framed area corresponding to the phases. The colors, in order of specification, are applied to the sub-rectangles starting from left to right. You can use the value *EMPTY* in the *color-list* to avoid filling a particular sub-rectangle. If the number of colors is less than the number of phases, the colors are applied cyclically. The colors are also used for phase legends requested with the PHASELEGEND option.

CFRAMELAB=*color*

specifies the color for filling rectangles that frame the point labels displayed with the ALLLABEL=, ALLLABEL2=, OUTLABEL=, and OUTLABEL2= options. By default, the points are not framed. *Graphics*

CGRID=*color*

specifies the color for the grid requested by the ENDGRID or GRID option. By default, the grid is the same color as the axes. *Graphics*

CHREF=*color*

Graphics

specifies the color for the lines requested by the HREF= and HREF2= options. The default is the first color in the device color list.

CIINDICES <(<**TYPE=***keyword*><**ALPHA=***value*>)>

requests capability index confidence limits based on subgroup summary data, calculated using "effective degrees of freedom" as described by Bissell (1990). These confidence limits are approximate. When you specify the CIINDICES option, the calculated confidence limits are available for display in an inset and are included in the OUTLIMITS= data set, if one is produced.

TYPE=*keyword*

specifies the type of confidence limit. Valid values are LOWER, UPPER and TWOSIDED. The default value is TWOSIDED.

ALPHA=*value*

specifies the default confidence level to compute confidence limits. The percentage for the confidence limits is $(1-value)*100$. For example, ALPHA=.05 results in a 95% confidence limit. The default value is .05 and the possible range of values is from 0 to 1.

CINFILL=*color*

Graphics

specifies the color for the area inside the upper and lower control limits. By default, this area is not filled with a color. See also the COUTFILL= option.

CLABEL=*color*

Graphics

specifies the color for labels produced by the ALLLABEL=, ALLLABEL2=, OUTLABEL=, and OUTLABEL2= options.

CLIMITS=*color*

Graphics

specifies the color for the control limits, the central line, and the labels for these lines. The default color is the first color in the device color list.

CLIPCHAR='*character***'**

Line Printer

specifies a plot character that identifies clipped points, as requested with the CLIPFACTOR= option. Specifying the CLIPCHAR= option is recommended when the CLIPFACTOR= option is used. The default character is an asterisk (*).

CLIPFACTOR=*factor*

requests clipping of extreme points on the control chart. The *factor* that you specify determines the extent to which these values are clipped, and it must be greater than one (useful values are in the range 1.5 to 2).

For examples of the CLIPFACTOR= option, see Figure 54.28 on page 1965 and Figure 54.29 on page 1966. The CLIPFACTOR= option should not be used in any statement in which the STARVERTICES= option is also used. Related clipping options are CCLIP=, CLIPCHAR=, CLIPLEGEND=, CLIPLEGPOS=, CLIPSUBCHAR=, and CLIPSYMBOL=.

CLIPLEGEND='*label***'**
specifies the *label* for the legend that indicates the number of clipped points when the CLIPFACTOR= option is used. The *label* must be no more than 16 characters and must be enclosed in quotes. For an example, see Figure 54.29 on page 1966.

CLIPLEGPOS=TOP | BOTTOM
specifies the position for the legend that indicates the number of clipped points when the CLIPFACTOR= option is used. The keywords TOP and BOTTOM position the legend at the top or bottom of the chart, respectively. Do not specify CLIPLEGPOS=TOP together with the PHASELEGEND option or the BLOCKPOS=1 or BLOCKPOS=2 options. By default, CLIPLEGPOS=BOTTOM.

CLIPSUBCHAR='*character***'**
specifies a substitution character (such as #) for the label provided with the CLIPLEGEND= option. The substitution character is replaced with the number of points that are clipped. For example, suppose that the following statements produce a chart in which three extreme points are clipped:

```
proc shewhart data=pistons;
   xrchart diameter*hour /
      clipfactor  = 1.5
      cliplegend  = 'Points clipped=#'
      clipsubchar = '#' ;
run;
```

Then the clipping legend displayed on the chart will be

```
Points clipped=3
```

CLIPSYMBOL=*symbol*
specifies a plot symbol used to identify clipped points on the chart and in the legend when the CLIPFACTOR= option is used. You should use this option in conjunction with the CLIPFACTOR= option. The default *symbol* is CLIPSYMBOL=SQUARE. *Graphics*

CLIPSYMBOLHT=*value*
specifies the height for the symbol marker used to identify clipped points on the chart when the CLIPFACTOR= option is used. The default is the height specified with the H= option in the SYMBOL statement. *Graphics*

For general information about clipping options, refer to "Clipping Extreme Points" on page 1962.

CNEEDLES=*color*
requests that points are to be connected to the central line with vertical line segments (needles) and specifies the color of the needles. You can use needles to visually represent the process as a series of shocks or vertical displacements away from a constant mean. See Figure 54.26 on page 1962 for an example. The default *color* is the second color in the device color list. The CNEEDLES= option is available in all chart statements except the BOXCHART statement. *Graphics*

CONNECTCHAR=*'character'*
CCHAR=*'character'*

Line Printer

specifies the character used to form line segments that connect points on a chart. The default character is a plus (+) sign.

CONTROLSTAT=MEAN | MEDIAN

specifies whether the control limits displayed in a box chart are to be computed for subgroup means or for subgroup medians. By default, CONTROLSTAT=MEAN. The CONTROLSTAT= option is available only in the BOXCHART statement.

COUT=*color*

Graphics

specifies the color for the plotting symbols and the portions of connecting line segments that lie outside the control limits. The default color is the second color in the device color list. This option is useful for highlighting out-of-control points.

COUTFILL=*color*

Graphics

specifies the fill color for the areas outside the control limits that lie between the connected points and the control limits and are bounded by connecting lines. This option is useful for highlighting out-of-control points. See Figure 56.11 on page 2014 for an example. By default, these areas are not filled. Note that you can use the CINFILL= option to fill the area inside the control limits.

COVERLAY=(*color-list***)**

Graphics

specifies the colors used to plot primary chart overlay variables. Colors in the COVERLAY= list are matched with variables in the corresponding positions in the OVERLAY= list.

COVERLAY2=(*color-list***)**

Graphics

specifies the colors used to plot secondary chart overlay variables. Colors in the COVERLAY2= list are matched with variables in the corresponding positions in the OVERLAY2= list.

COVERLAYCLIP=*color*

Graphics

specifies the color used to plot clipped values on overlay plots when the CLIPFACTOR= option is used.

CPHASEBOX=*color*

Graphics

specifies the color for a box that encloses all of the plotted points for a phase (group of consecutive observations that have the same value of the variable _PHASE_). By default, an enclosing box is not drawn. This option is available only in the BOXCHART statement.

CPHASEBOXCONNECT=*color*

Graphics

specifies the color for line segments that connect the vertical edges of adjacent enclosing boxes requested with the CPHASEBOX= option or the CPHASEBOXFILL= option. The vertical coordinates of the attachment points represent the average of the values plotted inside the box. The CPHASEBOXCONNECT= option is an alternative to the CPHASEMEANCONNECT= option. This option is available only in the BOXCHART statement.

CPHASEBOXFILL=*color*

specifies the fill color for a box that encloses all of the plotted points for a phase. By default, an enclosing box is not drawn. This option is available only in the BOXCHART statement. *Graphics*

CPHASELEG=*color*

specifies a text color for the phase labels requested with the PHASELEGEND option. By default, if you specify a list of fill colors with the CFRAME= option, these colors are used for the corresponding phase labels, otherwise, the CTEXT= color is used for the phase labels. *Graphics*

CPHASEMEANCONNECT=*color*

specifies the color for line segments that connect points representing the average of the values plotted within a phase. This option must be used in conjunction with the CPHASEBOX= or CPHASEBOXFILL= options, and it is an alternative to the CPHASEBOXCONNECT= option. The points are centered horizontally within the enclosing boxes. This option is available only in the BOXCHART statement. *Graphics*

CSTARCIRCLES=*color*

specifies a color for the circles requested with the STARCIRCLES= option. See "Displaying Auxiliary Data with Stars" on page 1948. By default, the color specified with the CSTARS= option is used. *Graphics*

CSTARFILL=*color*
CSTARFILL=(*variable***)**

specifies a color or colors for filling the interior of stars requested with the STARVERTICES= option. You can use one of the following approaches: *Graphics*

- Specify a single color to be used for all stars with CSTARFILL=*color*.
- Specify a distinct color for *each* star (or subsets of stars) by providing the colors as values of a variable specified with CSTARFILL=(*variable*). The variable must be a character variable of length 8 or less in the input data set, and its values must be valid SAS/GRAPH colors or the value *EMPTY*. The color for the star positioned at the i^{th} subgroup on the chart is the value of the CSTARFILL= *variable* in the observations corresponding to the i^{th} subgroup. Note that if there are multiple observations per subgroup in the input data set (for instance, if you are using the XRCHART statement in the SHEWHART procedure to analyze observations from a DATA= input data set), the values of the CSTARFILL= *variable* should be identical for all the observations in a given subgroup.

See "Displaying Auxiliary Data with Stars" on page 1948. By default, the interior of the stars is empty.

CSTAROUT=*color*

specifies a color for those portions of the outlines of stars (requested with the STARVERTICES= option) that exceed the inner or outer circles. This option applies only with the STARTYPE=RADIAL and STARTYPE=SPOKE options, and it is useful for highlighting extreme values of star vertex variables. See "Displaying Auxiliary Data with Stars" on page 1948. *Graphics*

CSTARS=*color*
CSTARS=(*variable***)**

Graphics

specifies a color or colors for the outlines of stars requested with the STARVERTICES= option.

You can use one of the following approaches:

- You can specify a single color to be used for all the stars on the chart with CSTARS=*color*.
- You can specify a distinct outline color for *each* star (or subsets of stars) by providing the colors as values of a variable specified with CSTARS=(*variable*). The variable must be a character variable of length 8 or less in the input data set. The outline color for the star positioned at the t^{th} subgroup on the chart is the value of the CSTARS=*variable* in the observations corresponding to the t^{th} subgroup. Note that if there are multiple observations per subgroup in the input data set (for instance, if you are using the XRCHART statement in the SHEWHART procedure to analyze observations from a DATA= input data set), the values of the CSTARS= *variable* should be identical for all the observations in a given subgroup.

See "Displaying Auxiliary Data with Stars" on page 1948. By default, the second color in the device color list is used.

CSYMBOL='*label***'**
CSYMBOL=C | CBAR | CPM | CPM2 | C0

specifies a label for the central line in a c chart. You can use the option in two ways:

- You can specify a quoted *label* of length 16 or less.
- You can specify one of the keywords listed in the following table. Each keyword requests a label of the form *symbol*=*value*, where *symbol* is the symbol given in the table, and *value* is the value of the central line. If the central line is not constant, only the symbol is displayed.

Keyword	Symbol Printed on Charts Produced by	
	Graphics Devices	Line Printers
C	C	C
CBAR	$\overline{\text{C}}$	$\overline{\text{C}}$
CPM	C′	C′
CPM2	C″	C″
C0	C_0	C0

See Example 40.2 on page 1340 for an example. The default keyword is CBAR. The CSYMBOL= option is available only in the CCHART statement.

CTESTLABBOX=*color*

specifies the color for boxes enclosing labels for positive tests for special causes requested with the TESTLABBOX option. If you use the CTESTLABBOX= option, you do not need to specify the TESTLABBOX option. *Graphics*

CTESTS=*color* **|** *test-color-list*
CTEST=*color* **|** *test-color-list*

specifies colors for labels indicating points where a test is positive. *Graphics*

- You can specify the *color* for the labels used to identify points at which tests for special causes specified in the TESTS= option are positive. For Tests 2 through 8, this color is also used for the line segments that connect patterns of points for which a test is positive. The default color is the second color in the device color list.
- You can specify the *test-color-list* to enable different colors to be used for the labels and highlighted line segments associated with different tests for special causes. Any positive tests for which no specific CTESTS= value is specified are displayed using the general CTESTS= color. A non-default general CTESTS= color can be specified using the CTESTS=*color* syntax.

The following options request the standard tests for special causes 1 through 4 and one user-defined test designated B.

```
TESTS  = 1 to 4 M(K=4 DIR=DEC Code=B);
CTESTS = green;
CTESTS = (1 purple 3 yellow B blue);
```

Test 1 will be displayed in purple, Test 3 in yellow, and Test B in blue. Tests 2 and 4 will be displayed in green, the general CTESTS= color.

CTESTSYMBOL=*color*
CTESTSYM=*color*

specifies the color of the symbol used to plot subgroups with positive tests for special causes. *Graphics*

CTEXT=*color*

specifies the color for tick mark values and axis labels. This color is also used for the sample size legend and for the control limit legend. The default color is the color specified in the CTEXT= option in the most recent GOPTIONS statement. *Graphics*

CVREF=*color*
CV=*color*

specifies the color for reference lines requested by the VREF= and VREF2= options. The default is the first color in the device color list. *Graphics*

CZONES=*color*

Graphics

requests lines marking zones A, B, and C for the tests for special causes (see the TESTS= option) and specifies the *color* for these lines. This color is also used for labels requested with the ZONELABELS option. The default color is the first color in the device color list.

DATAUNIT=PERCENT | PROPORTION

enables you to use proportions or percents as the values for *processes* when you are using the PCHART or NPCHART statements and reading a DATA= input data set. Specify DATAUNIT=PERCENT to indicate that the values are percents of nonconforming items. Specify DATAUNIT=PROPORTION to indicate that the values are proportions of nonconforming items. Values for percents can range from 0 to 100, while values for proportions can range from 0 to 1. By default, the values of *processes* read from a DATA= data set for PCHART and NPCHART statements are assumed to be numbers (counts) of nonconforming items. The DATAUNIT= option is available only in the NPCHART and PCHART statements.

DESCRIPTION='*string***'**
DES='*string***'**

Graphics

specifies a description for the primary chart of length 40 or less that appears in the PROC GREPLAY master menu. The default *string* is the variable name. A related option is NAME=.

DESCRIPTION2='*string***'**
DES2='*string***'**

Graphics

specifies a description for the secondary chart of length 40 or less that appears in the PROC GREPLAY master menu. The default *string* is the variable name. The DESCRIPTION2= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements, and it is used in conjunction with the SEPARATE option. A related option is NAME2=.

ENDGRID

adds a grid to the rightmost portion of the chart, beginning with the first labeled major tick mark position that follows the last plotted point. This grid is useful in situations where you want to add points by hand after the chart is created. You can use the HAXIS= option to force space to be added to the horizontal axis.

EXCHART

creates a control chart only when exceptions occur, specifically, when the control limits are exceeded or when any of the tests requested with the TESTS= option or the TESTS2= option are positive.

FONT=*font*

Graphics

specifies a software font for labels and legends. You can also specify fonts for axis labels in an AXIS statement. The FONT= font takes precedence over the FTEXT= font specified in the GOPTIONS statement. Hardware characters are used by default.

GRID

adds a grid to the control chart. Grid lines are horizontal lines positioned at labeled major tick marks, and they cover the length and height of the plotting area.

HAXIS=*values*
HAXIS=AXIS*n*

specifies tick mark values for the horizontal (subgroup) axis. If the subgroup variable is numeric, the *values* must be numeric and equally spaced. Numeric values can be given in an explicit or implicit list. If the subgroup variable is character, *values* must be quoted strings of length 32 or less. If a date, time, or datetime format is associated with a numeric subgroup variable, SAS datetime literals can be used. Examples of HAXIS= lists follow:

```
haxis=0 2 4 6 8 10
haxis=0 to 10 by 2
haxis='LT12A' 'LT12B' 'LT12C' 'LT15A' 'LT15B' 'LT15C'
haxis='20MAY88'D to '20AUG88'D by 7
haxis='01JAN88'D to '31DEC88'D by 30
```

If the subgroup variable is numeric, the HAXIS= list must span the subgroup variable values, and if the subgroup variable is character, the HAXIS= list must include all of the subgroup variable values. You can add subgroup positions to the chart by specifying HAXIS= values that are not subgroup variable values.

If you specify a large number of HAXIS= values, some of these may be thinned to avoid collisions between tick mark labels. To avoid thinning, use one of the following methods:

- Shorten values of the subgroup variable by eliminating redundant characters. For example, if your subgroup variable has values LOT1, LOT2, LOT3, and so on, you can use the SUBSTR function in a DATA step to eliminate "LOT" from each value, and you can modify the horizontal axis label to indicate that the values refer to lots.
- Use the TURNHLABELS option to turn the labels vertically.
- Use the NPANELPOS= option to force fewer subgroup positions per panel.

If you are using a graphics device, you can also specify a previously defined AXIS statement with the HAXIS= option.

HEIGHT=*value*

Graphics

specifies the height (in vertical screen percent units) of the text for axis labels and legends. This *value* takes precedence over the HTEXT= value specified in the GOPTIONS statement. This option is recommended for use with software fonts specified with the FONT= option or with the FTEXT= option in the GOPTIONS statement. Related options are LABELHEIGHT= and TESTHEIGHT=.

HMINOR=*n*
HM=*n*

Graphics

specifies the number of minor tick marks between each major tick mark on the horizontal axis. Minor tick marks are not labeled. The default is 0.

HOFFSET=*value*

Graphics

specifies the length in percent screen units of the offset at both ends of the horizontal axis. You can eliminate the offset by specifying HOFFSET=0.

HREF=*values*
HREF=*SAS-data-set*

draws reference lines perpendicular to the horizontal (subgroup) axis on the primary chart. You can use this option in the following ways:

- You can specify the *values* for the lines with an HREF= list. If the subgroup variable is numeric, the *values* must be numeric. If the subgroup variable is character, the *values* must be quoted strings of up to 32 characters. If the subgroup variable is formatted, the *values* must be given as internal values.

  Examples of HREF= *values* follow:

  ```
  href=5
  href=5 10 15 20 25 30
  href='Shift 1' 'Shift 2' 'Shift 3'
  ```

- You can specify the values for the lines as the values of a variable named _REF_ in an HREF= data set. The type and length of _REF_ must match those of the *subgroup variable* specified in the chart statement. Optionally, you can provide labels for the lines as values of a variable named _REFLAB_, which must be a character variable of length 16 or less. If you want distinct reference lines to be displayed in charts for different *processes* specified in the chart statement, you must include a character variable of length 32 or less named _VAR_, whose values are the *processes*. If you do not include the variable _VAR_, all of the lines are displayed in all of the charts.

  Each observation in the HREF= data set corresponds to a reference line. If BY variables are used in the input data set (DATA=, HISTORY=, or TABLE=), the same BY variable structure must be used in the HREF= data set unless you specify the NOBYREF option.

Related options are CHREF=, HREFCHAR=, HREFLABELS=, HREFLABPOS=, LHREF=, and NOBYREF.

HREF2=*values*
HREF2=*SAS-data-set*

draws reference lines perpendicular to the horizontal (subgroup) axis on the secondary chart. The conventions for specifying the HREF2= option are identical to those for specifying the HREF= option. Related options are CHREF=, HREFCHAR=, HREF2LABELS=, HREFLABPOS=, LHREF=, and NOBYREF. The HREF2= option is available only in the IRCHART, MRCHART, XRCHART,

and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option.

HREF2DATA=*SAS-data-set*
draws reference lines perpendicular to the horizontal (subgroup) axis on the secondary chart. The HREF2DATA= option must be used in place of the HREF2= option to specify a data set using the quoted filename notation.

HREF2LABELS='*label1*' . . . '*labeln*'
HREF2LABEL='*label1*' . . . '*labeln*'
HREF2LAB='*label1*' . . . '*labeln*'
specifies labels for the reference lines requested by the HREF2= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters. The HREF2LABELS= option is available only in the IRCHART, MRCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option.

HREFCHAR='*character*'
specifies the character used to form the reference lines requested by the HREF= and HREF2= options for a line printer. The default is the vertical bar (|). *Line Printer*

HREFDATA=*SAS-data-set*
draws reference lines perpendicular to the horizontal (subgroup) axis on the primary chart. The HREFDATA= option must be used in place of the HREF= option to specify a data set using the quoted filename notation.

HREFLABELS='*label1*' . . . '*labeln*'
HREFLABEL='*label1*' . . . '*labeln*'
HREFLAB='*label1*' . . . '*labeln*'
specifies labels for the reference lines requested by the HREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

HREFLABPOS=*n*
specifies the vertical position of the HREFLABEL= and HREF2LABEL= labels, as described in the following table. By default, n=2.

1	along top of subplot area
2	staggered from top to bottom of subplot area
3	along bottom of subplot area
4	staggered from bottom to top of subplot area

Figure 53.5 illustrates label positions for values of the HREFLABPOS= option when the VREF= and VREFLABELS= options are as follows:

```
href       = 2 4 7
hreflabels = 'Two' 'Four' 'Seven'
```

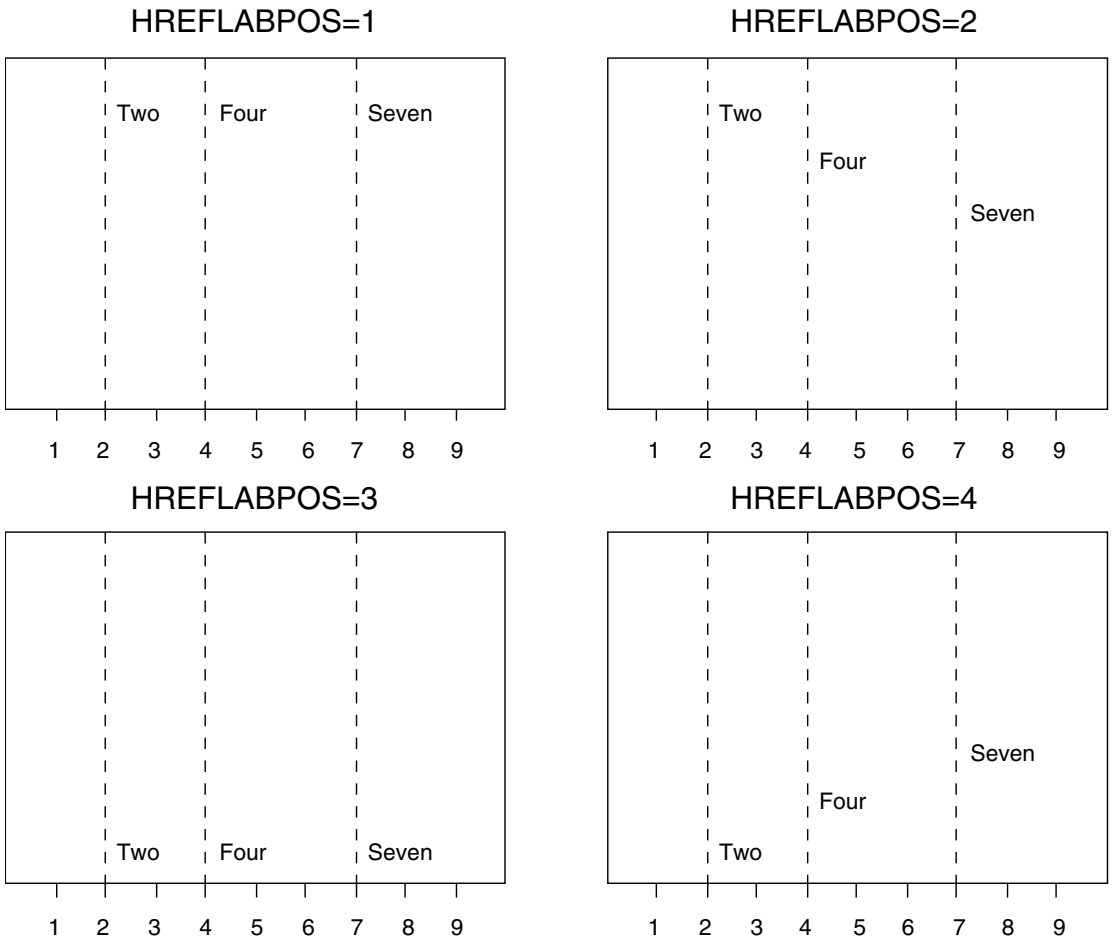


Figure 53.5. Positions for Reference Line Labels

HTML=*variable*

Graphics

specifies URLs as values of the specified character variable (or formatted values of a numeric variable). These URLs are associated with points on a primary control chart when high-resolution graphics output is directed into HTML. The value of the HTML= variable should be the same for each observation with a given value of the subgroup variable. See the chapter "Interactive Control Charts" for more information.

HTML2=*variable*

Graphics

specifies URLs as values of the specified character variable (or formatted values of a numeric variable). These URLs are associated with points on a secondary chart when high-resolution graphics output is directed into HTML. The value of the HTML= variable should be the same for each observation with a given value of the subgroup variable. See the chapter "Interactive Control Charts" for more information.

HTML_LEGEND=*variable*

Graphics

specifies URLs as values of the specified character variable (or formatted values of a numeric variable). These URLs are associated with symbols in the legend for the levels of a *symbol-variable*. The value of the HTML_LEGEND= variable should be the same for each observation with a given value of *symbol-variable*.

IDCOLOR=*color*

Graphics

specifies the color of the symbol marker used to identify outliers in schematic box-and-whisker plots produced with the BOXCHART statement when you use one of the following options: BOXSTYLE=SCHEMATIC, BOXSTYLE=SCHEMATICID, and BOXSTYLE=SCHEMATICIDFAR. The default *color* is the color specified with the CBOXES= option; otherwise, the second color in the device color list is used. The IDCOLOR option is available only in the BOXCHART statement.

IDCTEXT=*color*

Graphics

specifies the color for the text used to label outliers or indicate process variable values when you specify one of the keywords SCHEMATICID, SCHEMATICIDFAR, POINTSID, or POINTSJOINID with the BOXSTYLE= option. The default is the color specified with the CTEXT= option.

IDFONT=*font*

Graphics

specifies the font for the text used to label outliers or indicate process variable values when you specify one of the keywords SCHEMATICID, SCHEMATICIDFAR, POINTSID, or POINTSJOINID with the BOXSTYLE= option. The default *font* is SIMPLEX.

IDHEIGHT=*value*

Graphics

specifies the height for the text used to label outliers or indicate process variable values when you specify one of the keywords SCHEMATICID, SCHEMATICIDFAR, POINTSID, or POINTSJOINID with the BOXSTYLE= option. The default is the height specified with the HTEXT= option in the GOPTIONS statement.

IDSYMBOL=*symbol*

Graphics

specifies the symbol marker used to identify outliers in schematic box-and-whisker plots produced with the BOXCHART statement when you use one of the following options: BOXSTYLE=SCHEMATIC, BOXSTYLE=SCHEMATICID, and BOXSTYLE=SCHEMATICIDFAR. The default *symbol* is SQUARE. The IDSYMBOL= option is available only in the BOXCHART statement.

INTERVAL=DAY | DTDAY | HOUR | MINUTE | MONTH | QTR | SECOND

specifies the natural time interval between consecutive subgroup positions when a time, date, or datetime format is associated with a numeric subgroup variable. By default, the INTERVAL= option uses the number of subgroup positions per panel that you specify with the NPANELPOS= option. The default time interval keywords for various time formats are shown in the following table.

Format	Default Keyword	Format	Default Keyword
DATE	DAY	MONYY	MONTH
DATETIME	DTDAY	TIME	SECOND
DDMMYY	DAY	TOD	SECOND
HHMM	HOUR	WEEKDATE	DAY
HOUR	HOUR	WORDDATE	DAY
MMDDYY	DAY	YYMMDD	DAY
MMSS	MINUTE	YYQ	QTR

You can use the INTERVAL= option to modify the effect of the NPANELPOS= option, which specifies the number of subgroup positions per panel (screen or page). The INTERVAL= option enables you to match the scale of the horizontal axis to the scale of the subgroup variable without having to associate a different format with the subgroup variable.

For example, suppose your formatted subgroup values span an overall time interval of 100 days and a DATETIME format is associated with the subgroup variable. Since the default interval for the DATETIME format is DTDAY and since NPANELPOS=50 by default, the chart is displayed with two panels (screens or pages).

Now, suppose your data span an overall time interval of 100 hours and a DATETIME format is associated with the subgroup variable. The chart for these data are created in a single panel, but the data occupy only a small fraction of the chart since the scale of the data (hours) does not match that of the horizontal axis (days). If you specify INTERVAL=HOUR, the horizontal axis is scaled for 50 hours, matching the scale of the data, and the chart is displayed with two panels.

INTSTART=*value*

specifies the starting value for a numeric horizontal axis, when a date, time, or datetime format is associated with the subgroup variable. If the value specified is greater than the first subgroup variable value, this option has no effect.

LABELANGLE=*angle*

Graphics

specifies the angle at which labels requested with the ALLLABEL=, ALLLABEL2=, OUTLABEL=, and OUTLABEL2= options are drawn. A positive angle rotates the labels counterclockwise; a negative angle rotates them clockwise. By default, labels are oriented horizontally.

LABELFONT=*font*
TESTFONT=*font*

Graphics

specifies a software font for labels requested with the ALLLABEL=, ALLLABEL2=, OUTLABEL=, OUTLABEL2=, STARLABEL=, TESTLABEL=, and TESTLABELn= options. Hardware characters are used by default.

LABELHEIGHT=*value*
TESTHEIGHT=*value*

Graphics

specifies the height (in vertical percent screen units) for labels requested with the ALLLABEL=, ALLLABEL2=, OUTLABEL=, OUTLABEL2=, STARLABEL=, TESTLABEL=, and TESTLABELn= options. The default height is the height specified with the HEIGHT= option or the HTEXT= option in the GOPTIONS statement.

LBOXES=*linetype*
LBOXES=(*variable***)**

Graphics

specifies the line types for the outlines of the box-and-whisker plots created with the BOXCHART statement. You can use one of the following approaches:

- You can specify LBOXES=*linetype* to provide a single *linetype* for all of the box-and-whisker plots.
- You can specify LBOXES=(*variable*) to provide a distinct line type for *each* box-and-whisker plot. The *variable* must be a numeric variable in the input

data set, and its values must be valid SAS/GRAPH *linetype* values (numbers ranging from 1 to 46). The line type for the plot displayed for a particular subgroup is the value of the *variable* in the observations corresponding to this subgroup. Note that if there are multiple observations per subgroup in the input data set, the values of the *variable* should be identical for all of the observations in a given subgroup.

The default value is 1, which produces solid lines. The LBOXES= option is available only in the BOXCHART statement.

LCLLABEL=*'label'*
specifies a label for the lower control limit in the primary chart. The label can be of length 16 or less. Enclose the label in quotes. The default label is of the form *LCL=value* if the control limit has a fixed value; otherwise, the default label is *LCL*. Related options are LCLLABEL2=, UCLLABEL=, and UCLLABEL2=.

LCLLABEL2=*'label'*
specifies a label for the lower control limit in the secondary chart. The label can be of length 16 or less. Enclose the label in quotes. The default label is of the form *LCL=value* if the control limit has a fixed value; otherwise, the default label is *LCL*. The LCLLABEL2= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements. Related options are LCLLABEL=, UCLLABEL=, and UCLLABEL2=.

LENDGRID=*n*
specifies the line type for the grid requested with the ENDGRID option. The default is $n = 1$, which produces a solid line. If you use the LENDGRID= option, you do not need to specify the ENDGRID option. *Graphics*

LGRID=*n*
specifies the line type for the grid requested with the GRID option. The default is $n = 1$, which produces a solid line. If you use the LGRID= option, you do not need to specify the GRID option. *Graphics*

LHREF=*linetype*
LH=*linetype*
specifies the line type for reference lines requested with the HREF= and HREF2= options. The default is 2, which produces a dashed line. *Graphics*

LIMITN=*n*
LIMITN=VARYING
specifies either a fixed or varying nominal sample size for the control limits.

If you specify LIMITN=n, the control limits are computed for the fixed value n, and they do not vary with the subgroup sample sizes. Moreover, subgroup summary statistics are plotted *only* for those subgroups with a sample size equal to n. You can specify ALLN in conjunction with LIMITN=n to force all of the statistics to be plotted, regardless of subgroup sample size.

If you do not specify LIMITN=n and the subgroup sample sizes are constant, the default value of n is the constant subgroup sample size.

Depending on the chart statement, there are restrictions on the value of n that you can specify with the LIMITN= option. For the MRCHART, RCHART, and XRCHART statements, $2 \leq n \leq 25$. For the SCHART and XSCHART statements, $n \geq 2$. For the BOXCHART, MCHART, and XCHART statements, $n \geq 1$. If you omit the STDDEVIATIONS option for the MCHART or XCHART statements (or use the RANGES option with the BOXCHART statement) $n < 26$. For the CCHART and UCHART statements, $n > 0$, and n can assume fractional values (for all other chart statements, n must be a whole number). For the PCHART and NPCHART statements, $n \geq 1$.

For the IRCHART statement, n has a somewhat different interpretation; it specifies the number of consecutive measurements from which the moving ranges are to be computed, and $n \geq 2$. You can think of n as a *pseudo* nominal sample size for the control limits, since the data for an individual measurements and moving range chart are not subgrouped.

Note the difference between the LIMITN= option and the SUBGROUPN= option that is available in the CCHART, NPCHART, PCHART, and UCHART statements. The LIMITN= option specifies a nominal sample size for the *control limits*, whereas the SUBGROUPN= option provides the sample sizes for the *data*.

By default, LIMITN=2 in an IRCHART statement. You cannot specify LIMITN= VARYING in an IRCHART statement. For all other chart statements, LIMITN= VARYING is the default.

The following table identifies the chart features that vary when you use LIMITN= VARYING:

Chart Statement	Features Affected by LIMITN=VARYING
BOXCHART	control limits
CCHART	control limits, central line
MCHART	control limits
MRCHART	control limits on both charts, central line on R chart
NPCHART	control limits, central line
PCHART	control limits
RCHART	control limits, central line
SCHART	control limits, central line
UCHART	control limits
XCHART	control limits
XRCHART	control limits on both charts, central line on R chart
XSCHART	control limits on both charts, central line on s chart

Note: As an alternative to specifying the LIMITN= option, you can read the nominal control limit sample size from the variable _LIMITN_ in a LIMITS= data set. See "Input Data Sets" in the chapter for the chart statement in which you are interested.

LIMLABSUBCHAR='*character***'**
specifies a substitution character (such as #) for labels provided as quoted strings with the LCLLABEL=, LCLLABEL2=, UCLLABEL=, UCLLABEL2=, CSYMBOL=, NPSYMBOL=, PSYMBOL=, RSYMBOL=, SSYMBOL=, USYMBOL=, and XSYMBOL= options. The substitution character must appear in the label. When the label is displayed on the chart, the character is replaced with the value of the corresponding control limit or center line, provided that this value is constant across subgroups. Otherwise, the default label for a varying control limit or center line is displayed.

LLIMITS=*linetype*
specifies the line type for control limits. The default is 4, which produces a dashed line. *Graphics*

LOVERLAY=(*linetypes***)**
specifies line types for the line segments connecting points on primary chart overlays. Line types in the LOVERLAY= list are matched with variables in the corresponding positions in the OVERLAY= list. *Graphics*

LOVERLAY2=(*linetypes***)**
specifies line types for the line segments connecting points on secondary chart overlays. Line types in the LOVERLAY2= list are matched with variables in the corresponding positions in the OVERLAY2= list. *Graphics*

LSL=*value-list*
provides lower specification limits used to compute capability indices. If you provide more than one *value*, the number of *values* must match the number of *processes* listed in the chart statement. If you specify only one *value*, it is used for all the *processes*.

The SHEWHART procedure uses the specification limits to compute capability indices, and it saves the limits and indices in the OUTLIMITS= data set. For more information, see "Capability Indices" on page 1774 and "Output Data Sets" in the chapter for the chart statement in which you are interested. Also see the entry for the USL= option. The LSL= option is available in the BOXCHART, IRCHART, MCHART, MRCHART, RCHART, SCHART, XCHART, XRCHART, and XSCHART statements.

LSTARCIRCLES=*linetypes*
specifies one or more line types for the circles requested with the STARCIRCLES= option. The number of line types should match the number of circles requested, and the line types are paired with the circles in the order specified. The default *linetype* is 1, which produces a solid line. *Graphics*

Figure 53.6 illustrates circles displayed by the following LSTARCIRCLES= and STARCIRCLES= options:

```
starcircles  = 0.0 1.0 0.25 0.5
lstarcircles = 1 1 2 2
```

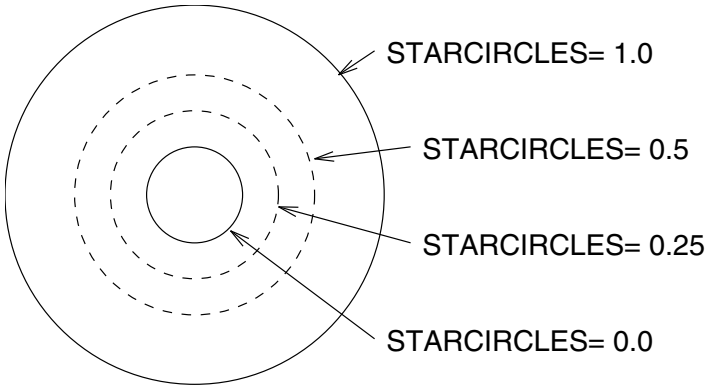


Figure 53.6. Line Types for Reference Circles

LSTARS=*linetype*
LSTARS=(*variable***)**

Graphics

specifies the line types for the outlines of stars requested with the STARVERTICES= option. You can use one of the following approaches:

- You can specify LSTARS=*linetype* to provide a single line type for all of the stars.
- You can specify LSTARS=(*variable*) to provide a distinct line type for *each* star. The variable must be a numeric variable in the input data set, and its values must be valid SAS/GRAPH line types. The line type for the star positioned at a particular subgroup is the value of the *variable* in the observations corresponding to this subgroup. Note that if there are multiple observations per subgroup in the input data set, the *variable* values should be identical for all of the observations in a given subgroup.

See "Displaying Auxiliary Data with Stars" on page 1948. The default *linetype* is 1, which produces a solid line.

LTESTS=*linetype*
LTEST=*linetype*

Graphics

specifies the line type for the line segments that connect patterns of points for which a test for special causes (requested with the TESTS= option) is positive. The default is 1, which produces a solid line.

LTMARGIN=*value*
LTM=*value*

Graphics

specifies the width (in horizontal percent screen units) of the left marginal area for the plot requested with the LTMPLOT= option. The LTMARGIN= option is available only in the IRCHART statement.

LTMPLOT=*keyword*

requests a univariate plot of the control chart statistics that is positioned in the left margin of the control chart. The *keywords* that you can specify and the associated plots are listed in the following table: *Graphics*

Keyword	Marginal Plot
HISTOGRAM	histogram
DIGIDOT	digidot plot
SKELETAL	skeletal box-and-whisker plot
SCHEMATIC	schematic box-and-whisker plot
SCHEMATICID	schematic box-and-whisker plot with outliers labeled
SCHEMATICIDFAR	schematic box-and-whisker plot with far outliers labeled

The LTMPLOT= option is available only in the IRCHART statement; see Example 41.3 on page 1386 for an example. Refer to Hunter (1988) for a description of digidot plots, and see the entry for the BOXSTYLE= option for a description of the various box-and-whisker plots. Related options are LTMARGIN=, RTMARGIN=, and RTMPLOT=.

LVREF=*linetype*
LV=*linetype*

specifies the line type for reference lines requested by the VREF= and VREF2= options. The default is 2, which produces a dashed line. *Graphics*

LZONES=*n*

specifies the line type for lines that delineate zones A, B, and C for standard tests requested with the TESTS= and/or TESTS2= options. The default is $n = 2$, which produces a dashed line. *Graphics*

MAXPANELS=*n*

specifies the maximum number of pages or screens for a chart. By default, $n = 20$.

MEDCENTRAL=AVGMEAN | AVGMED | MEDMED

identifies a method for estimating the process mean μ, which is represented by the central line on a median chart. The methods corresponding to each keyword are given in the following table:

Keyword	Method for Estimating Process Mean
AVGMEAN	average of subgroup means
AVGMED	average of subgroup medians
MEDMED	median of subgroup medians

The default keyword is AVGMED. The MEDCENTRAL= option is available only in the MCHART and MRCHART statements and in the BOXCHART statement with the CONTROLSTAT=MEDIAN option.

MISSBREAK

determines how subgroups are formed when observations are read from a DATA= data set and a character *subgroup-variable* is provided. When you specify the MISSBREAK option, observations with missing values of the *subgroup variable* are not processed. Furthermore, the next observation with a nonmissing value of the *subgroup-variable* is treated as the beginning observation of a new subgroup even if this value is identical to the most recent nonmissing subgroup value. In other words, by specifying the option MISSBREAK and by inserting an observation with a missing *subgroup-variable* value into a group of consecutive observations with the same *subgroup-variable* value, you can split the group into two distinct subgroups of observations.

By default, if MISSBREAK is not specified, observations with missing values of the *subgroup variable* are not processed, and all remaining observations with the same consecutive value of the *subgroup-variable* are treated as a single subgroup.

MRRESTART

MRRESTART=*value*

causes the moving range computation on the IRCHART to be restarted when a missing value is encountered. Without the MRRESTART option, a missing value is simply skipped, and the moving range for the next non-missing subgroup is computed using the most recent previous non-missing value. MRRESTART restarts the moving range computation, so only the observations after the missing value are used in subsequent moving range computations. MRRESTART restarts the moving range computation on any missing value; you can also specify MRRESTART=value to restart only on a particular missing value. For example, MRRESTART=R will restart the computation only when the missing value ".R" is encountered.

MU0=*value*

specifies a known (standard) value μ_0 for the process mean μ. By default, μ is estimated from the data. The MU0= option is available in the BOXCHART, IRCHART, MCHART, MRCHART, XCHART, XRCHART, and XSCHART statements.

Note: As an alternative to specifying MU0=μ_0, you can read a predetermined value for μ_0 from the variable _MEAN_ in a LIMITS= data set. See "Input Data Sets" in the chapter for the chart statement in which you are interested.

NAME='*string***'**

Graphics

specifies a name for the primary chart of length 8 or less that appears in the PROC GREPLAY master menu. The default name is 'SHEWHART'. A related option is DESCRIPTION=.

NAME2='*string***'**

Graphics

specifies a name for the secondary chart of length 8 or less that appears in the PROC GREPLAY master menu. The default name is 'SHEWHART'. The NAME2= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements, and it is used in conjunction with the SEPARATE option. A related option is DESCRIPTION2=.

NDECIMAL=*n*

specifies the number of decimal digits in the default labels for the control limits and the central line in the primary chart. The default is one more than the maximum number of decimal digits in the vertical axis tick mark labels. For example, if the vertical axis tick mark label with the largest number of digits after the decimal point is 110.05, the default is $n = 3$.

NDECIMAL2=*n*

specifies the number of decimal digits in the default labels for the control limits and central line in a secondary chart. The default is one more than the maximum number of decimal digits in the vertical axis tick mark labels. The NDECIMAL2= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements.

NEEDLES

connects plotted points to the central line with vertical line segments (needles). See Example 44.2 on page 1517 for an example. By default, adjacent points are connected to one another. The NEEDLES option is available in all chart statements except the BOXCHART statement.

NMARKERS

identifies a plotted subgroup summary statistic with a special symbol marker (character) when the corresponding subgroup sample size is not equal to the nominal control limit sample size n. Specify the nominal control limit sample size n with the LIMITN= option or with the variable _LIMITN_ read from a LIMITS= data set. The following table summarizes the identification:

Sample Size	Graphics Device Symbol	Line Printer Character
$< n$	$\bigtriangledown$	L
$> n$	$\bigtriangleup$	G

A legend that explains the symbols is displayed at the bottom of the chart. This legend can be suppressed with the NOLEGEND option.

The NMARKERS option is not available in the IRCHART statement. The NMARKERS option applies only when specified in conjunction with the ALLN option and a fixed nominal control limit sample size provided with the LIMITN= option or the variable _LIMITN_. See Example 50.3 on page 1781 for an illustration.

NO3SIGMACHECK

suppresses the check for 3σ limits when tests for special causes are requested. This enables tests for special causes to be applied when the SIGMAS= option is used to specify control limits other than the default 3σ limits. This option should not be used for standard control chart applications, since the standard tests for special causes assume 3σ limits.

NOBYREF

specifies that the reference line information in an HREF=, HREF2=, VREF=, or VREF2= data set is to be applied uniformly to charts created for all the BY groups in the input data set (DATA=, HISTORY=, or TABLE=). If you specify the NOBYREF

option, you do not need to provide BY variables in the reference line data set. By default, you must provide BY variables.

NOCHART
suppresses the creation of the chart. You typically specify the NOCHART option when you are using the procedure to compute control limits and save them in an output data set. You can also use the NOCHART option when you are tabulating results with the TABLE and related options.

In the IRCHART, MRCHART, XRCHART, and XSCHART statements, the NOCHART option suppresses the creation of both the primary and secondary charts. If you use a graphics device and specify the NOCHART option, the chart is not saved in a graphics catalog. To save the chart in a graphics catalog while suppressing the display of the chart, specify the NODISPLAY option in a GOPTIONS statement.

NOCHART2
suppresses the creation of a secondary chart. You typically use this option in the IRCHART statement to create a chart for individual measurements and suppress the accompanying chart for moving ranges. The NOCHART2 option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements.

NOCONNECT
suppresses line segments that connect points on the chart. By default, points are connected except in box charts produced with the BOXCHART statement (see the BOXCONNECT option).

NOCTL
suppresses the display of the central line in a primary chart.

NOCTL2
suppresses the display of the central line in a secondary chart. The NOCTL2 option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements.

NOFRAME
suppresses the default frame drawn around the chart.

NOHLABEL
suppresses the label for the horizontal (subgroup) axis. Use the NOHLABEL option when the meaning of the axis is evident from the tick mark labels, such as when a date format is associated with the subgroup variable.

NOLCL
suppresses the display of the lower control limit in a primary chart.

NOLCL2
suppresses the drawing of the lower control limit in a secondary chart. The NOLCL2 option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements.

NOLEGEND

suppresses the default legend for subgroup sample sizes, which appears by default below the chart. This option also suppresses the legend displayed by the NMARKERS option. Use the NOLEGEND option when the subgroup sample sizes are constant and equal to the control limit sample size, since the control limit sample size is displayed in the upper right corner of the chart.

NOLIMIT0

suppresses the display of a fixed lower control limit if and only if the value of the limit is zero. This option is useful in situations where a lower limit of zero is considered to be uninformative or visually distracting (for instance, on certain p charts or R charts). The NOLIMIT0 option is available with all chart statements except BOXCHART, MCHART, and XCHART. For the IRCHART, MRCHART, XRCHART, and XSCHART statements, the NOLIMIT0 option applies only to the secondary chart.

NOLIMIT1

suppresses the display of a fixed upper control limit on a p chart if and only if the value of the control limit is 1 (or 100%), or on an np chart if and only if the value of the control limit is n. The NOLIMIT1 option is available only in the NPCHART and PCHART statements.

NOLIMITLABEL

suppresses the default labels for the control limits and central lines.

NOLIMITS

suppresses the display of control limits. This option is particularly useful if you are using the BOXCHART statement to create side by side box-and-whisker plots; in this case, you should also use one of the BOXSTYLE= options.

NOLIMITSFRAME

suppresses the default frame for the control limit information that is displayed across the top of the chart when multiple sets of control limits with distinct multiples of σ and nominal control limit sample sizes are read from a LIMITS= data set.

NOLIMITSLEGEND

suppresses the legend for the control limits (for example, *3σ Limits For n=5*), which appears by default in the upper right corner of the chart.

NOOVERLAYLEGEND

suppresses the legend for overlay variables which is displayed by default when the OVERLAY= or OVERLAY2= option is specified. *Graphics*

NOPHASEFRAME

suppresses the default frame for the legend requested by the PHASELEGEND option.

NOREADLIMITS

specifies that the control limits for each *process* listed in the chart statement are *not* to be read from the LIMITS= data set specified in the PROC SHEWHART statement. There are two basic methods of displaying control limits: calculating control limits from the data and reading control limits from a LIMITS= data set. If you want control limits calculated from the data, you can do one of the following:

1. Do not specify a LIMITS= data set.
2. If you specify a LIMITS= data set, also specify the NOREADLIMITS option.

Otherwise, if you specify a LIMITS= data set in the PROC SHEWHART statement, the procedure reads control limits from that data set.*

The following example illustrates the NOREADLIMITS option:

```
proc shewhart data=pistons limits=diamlim;
   xrchart diameter*hour;
   xrchart diameter*hour / noreadlimits;
run;
```

The first XRCHART statement reads the control limits from the first observation in the data set DIAMLIM for which the variable _VAR_ is equal to `diameter` and the variable _SUBGRP_ is equal to `hour`. The second XRCHART statement computes the control limits from the measurements in the data set PISTONS. Note that the second XRCHART statement is equivalent to the following statements, which are more commonly used:

```
proc shewhart data=pistons;
   xrchart diameter*hour;
run;
```

For more information about reading control limits from a LIMITS= data set, see the entry for the READLIMITS option and "Displaying Multiple Sets of Control Limits" on page 1939.

NOTCHES

Graphics

specifies that box-and-whisker plots created by the BOXCHART statement are to be notched. The endpoints of the notches are located at the median plus and minus $1.58(\text{IQR}/\sqrt{n})$, where IQR is the interquartile range and n is the subgroup sample size. The medians (central lines) of two box-and-whisker plots are significantly different at approximately the 0.05 level if the corresponding notches do not overlap. Refer to McGill and others (1978). Figure 53.7 illustrates the NOTCHES option. Notice the folding effect at the bottom, which happens when the endpoint of a notch is beyond its corresponding quartile. This situation occurs typically only when the subgroup sample size is small.

*This is true for Release 6.10 and later releases of SAS/QC software. For Release 6.09 and earlier releases, the procedure calculates control limits from the data unless you specify a LIMITS= data set in the procedure statement **and** you specify either the READLIMITS option or the READINDEXES= option in the chart statement. The NOREADLIMITS option is not available for Release 6.09 and earlier releases. For more information, see the entry for the READLIMITS option.

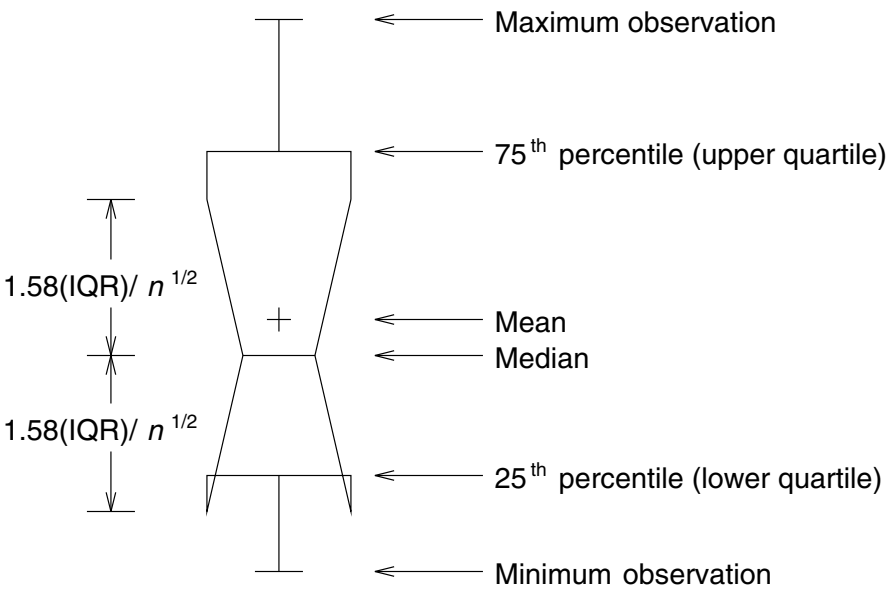


Figure 53.7. NOTCHES Option for Box-and-Whisker Plots

The NOTCHES option is also illustrated in Output 39.3.1 on page 1291 and is available only in the BOXCHART statement.

NOTICKREP
applies to character-valued *subgroup-variables* and specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on the horizontal axis.

NOTRENDCONNECT
suppresses line segments that connect points on a trend chart. Points are connected by default. The NOTRENDCONNECT option is available only in the BOXCHART, MCHART, and XCHART statements when the TRENDVAR= option is used.

NOTRUNC
overrides the vertical axis truncation at zero, which is applied by default to c charts, moving range charts, np charts, p charts, R charts, s charts, and u charts. This option is useful if you are creating a customized version of one of these charts and want to replace the plotted statistics and control limits with values read from a TABLE= input data set that can be positive or negative. Do not use the NOTRUNC option in standard control chart applications. This option is not available in the BOXCHART, MCHART, and XCHART statements.

NOUCL
suppresses the display of the upper control limit in a primary chart.

NOUCL2
suppresses the display of the upper control limit in a secondary chart. The NOUCL2 option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements.

NOVANGLE

Graphics

requests vertical axis labels that are strung out vertically. By default, the labels are drawn at an angle of 90 degrees if a software font is used.

NPANELPOS=*n*

NPANEL=*n*

specifies the number of subgroup positions per panel on each chart. A *panel* is defined as a screen or page (or a half-screen or half-page if you are also using the BILEVEL option). You typically specify the NPANELPOS= option to display more points on a panel than the default number, which is $n = 50$ for all chart statements except the BOXCHART statement, for which the default is $n = 20$.

You can specify a positive or negative number for n. The absolute value of n must be at least 5. If n is positive, the number of positions is adjusted so that it is approximately equal to n and so that all panels display approximately the same number of subgroup positions. If n is negative, no balancing is done, and each panel (except possibly the last) displays approximately $|n|$ positions. In this case, the approximation is due only to axis scaling.

You can use the INTERVAL= option to change the effect of the NPANELPOS= option when a date or time format is associated with the *subgroup-variable*. The INTERVAL= option enables you to match the scale of the horizontal axis to the scale of the subgroup variable without having to associate a different format with the subgroup variable.

NPSYMBOL='*label***'**

NPSYMBOL=NP | NPBAR | NPPM | NPPM2 | NP0

specifies a label for the central line in an np chart. You can use the option in the following ways:

- You can specify a quoted *label* up to 16 characters in length.
- You can specify one of the keywords listed in the following table. Each keyword requests a label of the form *symbol=value*, where *symbol* is one of the symbols given in the table, and *value* is the value of the central line. If the central line is not constant, only the symbol is displayed.

Keyword	Symbol Printed on Charts Produced by	
	Graphics Devices	Line Printers
NP	NP	NP
NPBAR	$\overline{\text{NP}}$	$\overline{\text{NP}}$
NPPM	NP′	NP′
NPPM2	NP″	NP″
NP0	NP_0	NP0

The default keyword is NPBAR. The NPSYMBOL= option is available only in the NPCHART statement.

OUTBOX=*SAS-data-set*

creates an output data set that contains subgroup summary statistics, control limits, and outlier values for a box chart. An OUTBOX= data set is the only type of summary data set produced by the SHEWHART procedure from which you can reconstruct a schematic box chart. The OUTBOX= option is available only in the BOXCHART statement. See "OUTBOX= Data Set" on page 1270 for details.

OUTHIGHHTML=*variable*

Graphics

specifies a variable whose values are URLs to be associated with outlier points above the upper fence on a schematic box chart when high-resolution graphics output is directed into HTML.

OUTHISTORY=*SAS-data-set*

creates an output data set that contains the subgroup summary statistics. You can use an OUTHISTORY= data set as a HISTORY= input data set in a subsequent run of the procedure. You cannot request an OUTHISTORY= data set if the input data set is a TABLE= data set. See "Output Data Sets" in the chapter for the chart statement in which you are interested. A related option is OUTPHASE=.

OUTINDEX='*label***'**

specifies the value of the _INDEX_ variable in the OUTLIMITS= output data set. This is a bookkeeping variable that provides information identifying the control limits saved in the data set. See "Output Data Sets" in the chapter for the chart statement in which you are interested.

The *label* can be up to 48 characters and should be enclosed in quotes. You should use a *label* that uniquely identifies the control limits. For example, you might specify OUTINDEX='April 1-15' to indicate that the limits were computed from data collected during the first half of April.

The OUTINDEX= option is intended to be used in conjunction with the OUTLIMITS= option. The _INDEX_ variable is created only if you specify the OUTINDEX= option. If you specify the OUTINDEX= option and do not specify the name of the OUTLIMITS= data set with the OUTLIMITS= option, the procedure creates an OUTLIMITS= data set whose name is of the form WORK.DATAn.

Note: You cannot use the OUTINDEX= and READINDEXES= options in the same chart statement.

OUTLABEL=VALUE
OUTLABEL=(*variable***)**

labels each point that falls outside the control limits on the primary chart with the VALUE plotted for that subgroup or with the value of a *variable* in the input data set.

The *variable* provided in the input data set can be numeric or character. If the *variable* is a character variable, it can be up to 16 characters. For each subgroup of observations whose summary statistic falls outside the control limits, the formatted value of the *variable* in the observations is used to label the point representing the subgroup. If you are reading a DATA= data set with multiple observations per subgroup, the values of the *variable* should be identical for observations within a subgroup. By default, points are not labeled. The OUTLABEL= option takes precedence over the

TESTLABEL= option when TESTS=1 is specified. You cannot specify both the OUTLABEL= and ALLLABEL= options.

OUTLABEL2=VALUE
OUTLABEL2=(*variable***)**
labels each point that falls outside the control limits on an R or s chart with the VALUE plotted for that subgroup or with the value of a *variable* in the input data set.

The *variable* provided in the input data set can be numeric or character. If the *variable* is a character variable, its length cannot exceed 16. For each subgroup of observations whose summary statistic falls outside the control limits, the formatted value of the *variable* in the observations is used to label the point representing the subgroup. If you are reading a DATA= data set with multiple observations per subgroup, the values of the *variable* should be identical for observations within a subgroup. By default, points are not labeled. The OUTLABEL2= option takes precedence over the TESTLABEL2= option when TESTS2=1 is specified. You cannot specify both the OUTLABEL2= and ALLLABEL2= options. The OUTLABEL2= option is available only in the IRCHART, MRCHART, RCHART, SCHART, XRCHART, and XSCHART statements.

OUTLIMITS=*SAS-data-set*
creates an output data set that saves the control limits. You can use an OUTLIMITS= data set as an input LIMITS= data set in a subsequent run of the procedure. See "Output Data Sets" in the chapter for the chart statement in which you are interested. A related option is OUTINDEX=.

OUTLOWHTML=*variable*
Graphics
specifies a variable whose values are URLs to be associated with outlier points below the lower fence on a schematic box chart when high-resolution graphics output is directed into HTML.

OUTPHASE='*label***'**
specifies the value of the _PHASE_ variable in the OUTHISTORY= data set. This is a bookkeeping variable that provides information identifying the summary statistics saved in the data set. See "Output Data Sets" in the chapter for the chart statement in which you are interested.

You should use the OUTPHASE= option if you create OUTHISTORY= data sets at different stages (phases) for the same *processes* and concatenate the data sets to build a master historical data set. The _PHASE_ variable then identifies the block of observations that corresponds to each phase.

The *label* can be up to 48 characters and should be enclosed in quotes. You should use a *label* that uniquely identifies the saved data. For example, you might specify OUTPHASE='April 1-15' to indicate that the data were collected during the first half of April.

The _PHASE_ variable is created only if you specify the OUTPHASE= option. If you specify the OUTPHASE= option and do not specify the name of the OUTHISTORY= data set with the OUTHISTORY= option, the procedure creates an OUTHISTORY= data set whose name is of the form WORK.DATAn.

OUTTABLE=*SAS-data-set*
creates an output SAS data set that saves the information plotted on the chart, including the subgroup variable values and their corresponding summary statistics and control limits.

You can use the OUTTABLE= data set to create a customized report with the reporting procedures and methods described in *SAS Language Reference: Dictionary*. You can also use an OUTTABLE= data set as a TABLE= input data set in a subsequent run of the procedure. See "Output Data Sets" in the chapter for the chart statement in which you are interested.

OVERLAY=(*variable-list***)**
specifies variables to be overlayed on the primary control chart. A point is plotted for each overlay variable at each subgroup for which it has a non-missing value. The value of a particular overlay variable should be the same for each observation in the input data set with a given value of the subgroup variable. If values differ within a subgroup, the first value appearing in that subgroup is used. The OVERLAY= option cannot be specified with the STARVERTICES= option. *Graphics*

OVERLAY2=(*variable-list***)**
specifies variables to be overlayed on a secondary control chart. A point is plotted for each overlay variable at each subgroup for which it has a non-missing value. The value of a particular overlay variable should be the same for each observation in the input data set with a given value of the subgroup variable. If values differ within a subgroup, the first value appearing in that subgroup is used. The OVERLAY2= option cannot be specified with the STARVERTICES= option. *Graphics*

OVERLAY2HTML=(*variable-list***)**
specifies variables whose values are URLs to be associated with points on secondary chart overlays. These URLs are associated with points on an overlay plot when high-resolution graphics output is directed into HTML. Variables in the OVERLAY2HTML= list are matched with variables in the corresponding positions in the OVERLAY2= list. The value of the OVERLAY2HTML= variable should be the same for each observation with a given value of the subgroup variable. *Graphics*

OVERLAY2ID=(*variable-list***)**
specifies variables whose formatted values are used to label points on secondary chart overlays. Variables in the OVERLAY2ID= list are matched with variables in the corresponding positions in the OVERLAY2= list. The value of the OVERLAY2ID= variable should be the same for each observation with a given value of the subgroup variable. *Graphics*

OVERLAY2SYM=(*symbol-list***)**
specifies symbols used to plot overlays on a secondary control chart. Symbols in the OVERLAY2SYM= list are matched with variables in the corresponding positions in the OVERLAY2= list. *Graphics*

OVERLAY2SYMHT=(*value-list***)**

Graphics

specifies the heights of symbols used to plot overlays on a secondary control chart. Heights in the OVERLAY2SYMHT= list are matched with variables in the corresponding positions in the OVERLAY2= list.

OVERLAYCLIPSYM=*symbol*

Graphics

specifies the symbol used to plot clipped values on overlay plots when the CLIPFACTOR= option is used.

OVERLAYCLIPSYMHT=*value*

Graphics

specifies the height for the symbol used to plot clipped values on overlay plots when the CLIPFACTOR= option is used.

OVERLAYHTML=(*variable-list***)**

Graphics

specifies variables whose values are URLs to be associated with points on primary chart overlays. These URLs are associated with points on an overlay plot when high-resolution graphics output is directed into HTML. Variables in the OVERLAYHTML= list are matched with variables in the corresponding positions in the OVERLAY= list. The value of the OVERLAYHTML= variable should be the same for each observation with a given value of the subgroup variable.

OVERLAYID=(*variable-list***)**

Graphics

specifies variables whose formatted values are used to label points on primary chart overlays. Variables in the OVERLAYID= list are matched with variables in the corresponding positions in the OVERLAY= list. The value of the OVERLAYID= variable should be the same for each observation with a given value of the subgroup variable.

OVERLAYLEGLAB='*label***'**

Graphics

specifies the label displayed to the left of the legend for overlays requested with the OVERLAY= or OVERLAY2= option. The label can be up to 16 characters and must be enclosed in quotes.

OVERLAYSYM=(*symbol-list***)**

Graphics

specifies symbols used to plot overlays on the primary control chart. Symbols in the OVERLAYSYM= list are matched with variables in the corresponding positions in the OVERLAY= list.

OVERLAYSYMHT=(*value-list***)**

Graphics

specifies the heights of symbols used to plot overlays on the primary control chart. Heights in the OVERLAYSYMHT= list are matched with variables in the corresponding positions in the OVERLAY= list.

P0=*value*

specifies a known (standard) value p_0 for the proportion of nonconforming items produced by the process. By default, p_0 is estimated from the data. The P0= option is available only in the NPCHART and PCHART statements.

Note: As an alternative to specifying P0=p_0, you can read a predetermined value for p_0 from the variable _P_ in a LIMITS= data set. See "Input Data Sets" in the chapter for the chart statement in which you are interested.

PAGENUM='*string***'**

specifies the form of the label used for pagination. *Graphics*

The *string* must be no longer than 16 characters, and it must include one or two occurrences of the substitution character #. The first # is replaced with the page number, and the optional second # is replaced with the total number of pages.

The PAGENUM= option is useful when you are working with a large number of subgroups, resulting in multiple pages of output. For example, suppose that each of the following XRCHART statements produces multiple pages:

```
proc shewhart data=pistons;
   xrchart diameter*hour / pagenum='Page #';
   xrchart diameter*hour / pagenum='Page # of #';
   xrchart diameter*hour / pagenum='#/#';
run;
```

The third page produced by the first statement would be labeled *Page 3*. The third page produced by the second statement would be labeled *Page 3 of 5*. The third page produced by the third statement would be labeled *3/5*.

By default, no page number is displayed.

PAGENUMPOS=TL | TR | BL | BR | TL100 | TR100 | BL0 | BR0

specifies where to position the page number requested with the PAGENUM= option. The keywords TL, TR, BL, and BR correspond to the positions top left, top right, bottom left, and bottom right, respectively. You can use the TL100 and TR100 keywords to ensure that the page number appears at the very top of a page when a title is displayed. The BL0 and BR0 keywords ensure that the page number appears at the very bottom of a page when footnotes are displayed. The default keyword is BR. *Graphics*

PCTLDEF=*index*

specifies one of five definitions used to calculate percentiles in the construction of box-and-whisker plots requested with the BOXCHART statement. The *index* can be 1, 2, 3, 4, or 5. The five corresponding percentile definitions are discussed in "Percentile Definitions" on page 1282 in Chapter 39, "BOXCHART Statement." The default *index* is 5. The PCTLDEF= option is available only in the BOXCHART statement.

PHASEBREAK

specifies that the last point in a phase (defined as a block of consecutive subgroups with the same value of the _PHASE_ variable) is not to be connected to the first point in the next phase. By default, the points are connected.

PHASELABTYPE=SCALED | TRUNCATED
PHASELABTYPE=*height*

specifies how lengthy _PHASE_ variable values are to be displayed when there is insufficient space in the legend requested with the PHASELEGEND option. *Graphics*

If you specify PHASELABTYPE=SCALED, the values are uniformly reduced in height so that they fit. If you specify PHASELABTYPE=TRUNCATED, lengthy values are truncated on the right until they fit. You can also specify a text *height*

in vertical percent screen units for the values. By default, lengthy values are not displayed. Related options are PHASELEGEND and PHASEREF.

PHASELEGEND
PHASELEG
identifies the phases requested with the READPHASES= option in a legend across the top of the chart. Related options are PHASELABTYPE= and PHASEREF.

PHASELIMITS
specifies that the control limits and center line are to be labeled for each phase specified with the READPHASES= option, providing the limits are constant within that phase.

PHASEMEANSYMBOL=*symbol*
Graphics
specifies a symbol marker for the average of the values plotted within a phase. This option is available only in the BOXCHART statement.

PHASEREF
delineates the phases specified with the READPHASES= option with reference lines drawn vertically. Related options are PHASELABTYPE= and PHASELEGEND.

PHASEVARLABEL
displays the label associated with the variable _PHASE_ above the phase values in the phase legend. If there is no label associated with _PHASE_, or if the PHASELEGEND option is not specified, PHASEVARLABEL has no effect.

PHASEVALSEP
displays vertical lines separating phase values in the phase legend. If the PHASELEGEND option is not specified, PHASEVALSEP has no effect.

POINTSHTML=*variable*
Graphics
specifies a variable whose values are URLs to be associated with points on a box chart when the BOXSTYLE= value is POINTS, POINTSJOIN, POINTSBOX, POINTSID, or POINTSJOINID. These URLs are associated with points on a box chart when graphics output is directed into HTML.

PSYMBOL='*label***'**
PSYMBOL=P | PBAR | PPM | PPM2 | P0
specifies a label for the central line in a *p* chart. You can use the option in the following ways:

- Specify a quoted *label* up to 16 characters.
- Specify one of the keywords listed in the following table. Each keyword requests a label of the form *symbol*=*value*, where *symbol* is the symbol given in the table, and *value* is the value of the central line. If the central line is not constant, only the symbol is displayed.

Keyword	Symbol Printed on Charts Produced by	
	Graphics Devices	Line Printers
P	P	P
PBAR	$\overline{P}$	$\overline{P}$
PPM	P'	P'
PPM2	P''	P''
P0	P_0	P0

The default keyword is PBAR. The PSYMBOL= option is available only in the PCHART statement.

RANGES
estimates the process standard deviation for a boxplot using subgroup ranges. By default, the process standard deviation for a boxplot is estimated from the subgroup standard deviations.

READALPHA
specifies that the variable _ALPHA_, rather than the variable _SIGMAS_, is to be read from a LIMITS= data set when both variables are available in the data set. Thus, the limits displayed are probability limits. If you do not specify the READALPHA option, then _SIGMAS_ is read by default. For details, see "Input Data Sets" in the chapter for the chart statement in which you are interested.

READINDEX=*value-list* **| ALL**
READINDEXES=*value-list* **| ALL**
READINDICES=*value-list* **| ALL**
reads one or more sets of control limits from a LIMITS= data set (specified in the PROC SHEWHART statement) for each *process* listed in the chart statement. The t^{th} set of control limits for a particular *process* is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches *process*
- the value of _SUBGRP_ matches the *subgroup variable*
- the value of _INDEX_ matches *value*

The *values* can be up to 48 characters and must be enclosed in quotes.

Note: You cannot use the READINDEX= and OUTINDEX= options in the same chart statement. Also, the READLIMITS and READINDEX= options are alternatives to each other. If the LIMITS= data set contains more than one set of control limits for the same *process*, you should use the READINDEX= option.

You can display distinct sets of control limits (read from a LIMITS= data set) with data for various *phases* (read from blocks of observations in the input data set) by using the READINDEXES= and READPHASES= options together. See the entry for the READPHASES= option.

For more information about multiple sets of control limits and about the keyword ALL, see "Displaying Multiple Sets of Control Limits" on page 1939.

READLIMITS

specifies that the control limits are to be read from a LIMITS= data set specified in the PROC SHEWHART statement.* The control limits for each *process* listed in the chart statement are to be read from the first observation in the LIMITS= data set where

- the value of _VAR_ matches *process*
- the value of _SUBGRP_ matches the *subgroup variable*

The use of the READLIMITS option depends on the release of SAS/QC software that you are using.

- **In Release 6.10 and later releases, the READLIMITS option is not necessary.** To read control limits as described previously, you simply specify a LIMITS= data set. However, even though the READLIMITS option is redundant, it continues to function as in earlier releases. Consequently, the following two XRCHART statements are equivalent:

```
proc shewhart data=pistons limits=diamlim;
   xrchart diameter*hour;
   xrchart diameter*hour / readlimits;
run;
```

  If the LIMITS= data set contains more than one set of control limits for the same *process*, you should use the READINDEX= option.

- **In Release 6.09 and earlier releases, you must specify the READLIMITS option to read control limits as described previously.** If you specify a LIMITS= data set without specifying the READLIMITS option (or the READINDEX= option), the control limits are computed from the data. Consequently, the following two XRCHART statements are **not** equivalent:

```
proc shewhart data=pistons limits=diamlim;
   xrchart diameter*hour;  /* limits computed from data */
   xrchart diameter*hour /
      readlimits;          /* limits read from DIAMLIM  */
   run;
```

  The READLIMITS and READINDEX= options are alternatives to each other.

  You can use the READLIMITS and READPHASES= options together. In this case, the control limits are read as described previously, and the data plotted on the chart are those selected by the READPHASES= option.

*For details about computing control limits from the data, see the entry for the NOREADLIMITS option on page 1885.

READPHASES=*value-list* **| ALL**
READPHASE=*value-list* **| ALL**

selects blocks of consecutive observations to be read from the input data set. You can use the READPHASES= option only if

- the input data set contains a _PHASE_ variable
- the _PHASE_ variable is a character variable of no more than 48 characters

The READPHASES= option selects those observations whose _PHASE_ value matches one of the *values* specified in the *value-list*. The block of consecutive observations identified by the i^{th} *value* is referred to as the i^{th} *phase*. The *values* can be up to 48 characters and must be enclosed in quotes. List the *values* in the same order that they appear as values of the variable _PHASE_ in the input data set.

With the READPHASES= option you can

- create control charts that label blocks of data corresponding to multiple time *phases*. See the PHASELEGEND, PHASEREF, and CFRAME= options.
- create *historical control charts* that display distinct sets of control limits for different *phases*. This also requires a LIMITS= data set and the READINDEXES= option.

If the subgroup variable is numeric, the values of the subgroup variable should be contiguous from one block of observations to the next. Otherwise, there may be a gap in the control chart between the last point in one phase and the first point in the next phase. If you read a data set that contains multiple observations for each subgroup, the value of _PHASE_ must be constant within the subgroup.

You can display distinct sets of control limits (read from a LIMITS= data set) with data for various *phases* by using the READINDEX= and READPHASES= options together. For example, consider the flange width data in the HISTORY= data set FLANGE and the LIMITS= data set FLANLIM. A partial listing of FLANGE is given in Figure 53.8 (for a complete listing of FLANGE, see Figure 54.7 on page 1940). The complete listing of FLANLIM is given in Figure 53.9.

Obs	_phase_	day	sample	flwidthx	flwidthr	flwidthn
1	Production	08FEB90	6	0.97360	0.06247	5
2	Production	09FEB90	7	1.00486	0.11478	5
3	Production	10FEB90	8	1.00251	0.13537	5
.	.	.	.	.	.	.
.	.	.	.	.	.	.
10	Production	19FEB90	15	0.99604	0.08242	5
11	Change 1	22FEB90	16	0.99218	0.09787	5
12	Change 1	23FEB90	17	0.99526	0.02017	5
.	.	.	.	.	.	.
.	.	.	.	.	.	.
20	Change 1	05MAR90	25	1.00412	0.04815	5
21	Change 2	08MAR90	26	1.00261	0.05604	5
22	Change 2	09MAR90	27	0.99553	0.02818	5
.	.	.	.	.	.	.
.	.	.	.	.	.	.
30	Change 2	19MAR90	35	1.00863	0.02649	5

Figure 53.8. Listing of the HISTORY= Data Set FLANGE

Obs	_index_	_var_	_subgrp_	_type_	_limitn_	_alpha_	_sigmas_
1	Change 1	FLWIDTH	SAMPLE	ESTIMATE	5	.0026998	3
2	Production	FLWIDTH	SAMPLE	ESTIMATE	5	.0026998	3
3	Start	FLWIDTH	SAMPLE	ESTIMATE	5	.0026998	3

Obs	_lclx_	_mean_	_uclx_	_lclr_	_r_	_uclr_	_stddev_
1	0.96167	0.99924	1.03680	0	0.06513	0.13771	0.028000
2	0.93792	0.98827	1.03862	0	0.08729	0.18458	0.037530
3	0.87088	0.96803	1.06517	0	0.16842	0.35612	0.072409

Figure 53.9. Listing of the LIMITS= Data Set FLANLIM

The following statements use the READINDEX= and READPHASES= options to create a historical control chart for the *Production* and *Change 1* phases:

```
proc shewhart history=flange limits=flanlim;
  xchart flwidth*sample /
     readphases  = ('Production' 'Change 1')
     readindexes = ('Production' 'Change 1')
     phaseref
     phaselegend ;
run;
```

The chart is displayed in Figure 53.10.

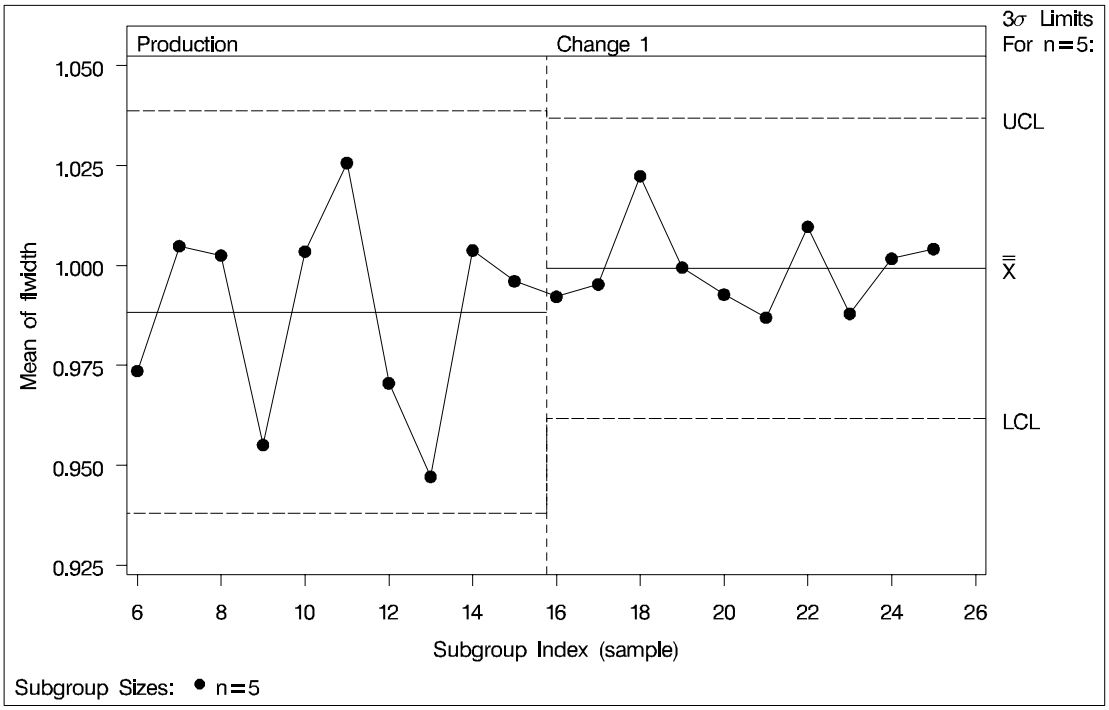


Figure 53.10. Multiple Control Limits for Multiple Phases

You can also use the keyword ALL with the READPHASES= option to match control limits to phases. For more information and examples about specifying multiple control limits, including the use of the keyword ALL, see "Displaying Multiple Sets of Control Limits" on page 1939.

REPEAT
REP

specifies that the horizontal axis of a chart that spans multiple pages is to be arranged so that the last subgroup position on a page is repeated as the first subgroup position on the next page. The REPEAT option facilitates cutting and pasting panels together. If a SAS DATETIME format is associated with the subgroup variable, REPEAT is used by default.

RSYMBOL='*label***'**
RSYMBOL=R | RBAR | RPM | R0

specifies a label for the central line in an R chart. You can use the option in the following ways:

- You can specify a quoted *label* up to 16 characters.
- You can specify one of the keywords listed in the following table. Each keyword requests a label of the form *symbol*=*value*, where *symbol* is the symbol given in the table, and *value* is the value of the central line. If the central line is not constant, only the symbol is displayed.

Keyword	Symbol Printed on Charts Produced by	
	Graphics Devices	Line Printers
R	R	R
RBAR	$\overline{R}$	$\overline{R}$
RPM	R'	R'
R0	R_0	R0

The default keyword is RBAR. The RSYMBOL= option is available only in the IRCHART, MRCHART, RCHART, and XRCHART statements.

RTMARGIN=*value*
RTM=*value* *Graphics*

specifies the width (in horizontal percent screen units) of the right marginal area for the plot requested with the RTMPLOT= option. The RTMARGIN= option is available only in the IRCHART statement.

RTMPLOT=*keyword*

requests a univariate plot of the control chart statistics that is positioned in the right margin of the control chart. The *keywords* that you can specify and the associated plots are listed in the following table: *Graphics*

Keyword	Marginal Plot
DIGIDOT	digidot plot
HISTOGRAM	histogram
SKELETAL	skeletal box-and-whisker plot
SCHEMATIC	schematic box-and-whisker plot
SCHEMATICID	schematic box-and-whisker plot with outliers labeled
SCHEMATICIDFAR	schematic box-and-whisker plot with far outliers labeled

The RTMPLOT= option is available only in the IRCHART statement; see Example 41.3 on page 1386 for an example. Refer to Hunter (1988) for a description of digidot plots, and see the entry for the BOXSTYLE= option for a description of the various box-and-whisker plots. Related options are LTMARGIN=, LTMPLOT=, and RTMARGIN=.

SEPARATE
displays primary and secondary charts on separate screens or pages. This option is useful if you are displaying line printer output on a terminal and the number of lines on the screen limits the resolution of the chart. The SEPARATE option is available only in the IRCHART, MRCHART, XRCHART, and XSCHART statements.

SERIFS
adds serifs to the whiskers of *skeletal box-and-whisker charts*. The SERIFS option is available only in the BOXCHART statement.

SIGMA0=*value*
specifies a known (standard) value σ_0 for the process standard deviation σ. By default, σ_0 is estimated from the data.

The SIGMA0= option is available in the BOXCHART, IRCHART, MCHART, MRCHART, RCHART, SCHART, XCHART, XRCHART, and XSCHART statements.

Note: As an alternative to specifying SIGMA0=σ_0, you can read a predetermined value for σ_0 from the variable _STDDEV_ in a LIMITS= data set. For details, see "Input Data Sets" in the chapter for the chart statement in which you are interested.

SIGMAS=k
specifies the width of the control limits in terms of the multiple k of the standard error of the subgroup summary statistic plotted on the chart. The value of k must be positive. By default, $k = 3$ and the control limits are "3σ limits."

The particular subgroup summary statistic whose standard error is multiplied by k depends on the chart statement, as indicated by the following table:

Statement	Subgroup Summary Statistic
BOXCHART	mean or median
CCHART	number nonconforming
IRCHART	individual measurements and moving ranges
MCHART	median
MRCHART	median and range
NPCHART	number nonconforming
PCHART	proportion nonconforming
RCHART	range
SCHART	standard deviation
UCHART	number of nonconformities per unit
XCHART	mean
XRCHART	mean and range
XSCHART	mean and standard deviation

For details, see the Options for Specifying Control Limits table and the "Details" section in the chapter for the particular chart statement that you are using.

Note that

- as an alternative to specifying SIGMAS=k, you can read k from the variable _SIGMAS_ in a LIMITS= data set. For details, see "Input Data Sets" in the chapter for the chart statement in which you are interested.
- as an alternative to specifying SIGMAS=k (or reading _SIGMAS_ from a LIMITS= data set), you can request probability limits by specifying ALPHA=α (or reading the variable _ALPHA_ from a LIMITS= data set by specifying the READALPHA option).

SKIPHLABELS=n
SKIPHLABEL=n

specifies the number n of consecutive tick mark labels, beginning with the second tick mark label, that are thinned (not displayed) on the horizontal (subgroup) axis. For example, specifying SKIPHLABEL=1 causes every other label to be skipped (not displayed). Specifying SKIPHLABEL=2 causes the second and third labels to be skipped, the fifth and sixth labels to be skipped, and so forth.

The default value of the SKIPHLABELS= option is the smallest value n for which tick mark labels do not collide. A specified n will be overridden to avoid collision, unless you specify SKIPHLABELS=0, which forces all tick mark labels to be displayed. To avoid both collisions and thinning, you can use the TURNHLABELS option.

SMETHOD=NOWEIGHT | MVLUE | RMSDF | MAD | MMR | MVGRANGE

specifies a method for estimating the process standard deviation, σ, as summarized by the following table:

Keyword	Method for Estimating Standard Deviation
NOWEIGHT	estimates σ as an unweighted average of unbiased subgroup estimates of σ
MVLUE	calculates a minimum variance linear unbiased estimate for σ
RMSDF	calculates a root-mean square estimate for σ
MAD	calculates a median absolute deviation estimate for σ (IRCHART only)
MMR	calculates a median moving range estimate for σ (IRCHART only)
MVGRANGE	estimates σ based on a moving range of subgroup means (XRCHART and XSCHART only)

For formulas, see "Methods for Estimating the Process Standard Deviation" in the chapter for the particular chart statement you are using.

The default keyword is NOWEIGHT. The SMETHOD= option is available in the BOXCHART, IRCHART, MCHART, MRCHART, RCHART, SCHART, XCHART, XRCHART, and XSCHART statements. You can specify SMETHOD=RMSDF only in the BOXCHART, MCHART, XCHART, SCHART, and XSCHART statements and only when used with the STDDEVIATIONS option (or only in the absence of the RANGES option with a BOXCHART statement). You can specify SMETHOD=MAD and SMETHOD=MMR only in the IRCHART statement. You can specify SMETHOD=MVGRANGE only in the XRCHART and XSCHART statements.

SPLIT='*character***'**

specifies a special *character* that is inserted into the label of a process variable or summary statistic variable and whose purpose is to split the label into two parts. The first part is used to label the vertical axis of the primary chart, and the second part is used to label the vertical axis of the secondary chart. The *character* is not displayed in either label. See Figure 54.31 on page 1969 for an example.

The SPLIT= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option.

SSYMBOL='*label***'**
SSYMBOL=S | SBAR | SPM | S0

specifies a label for the central line in an s chart. You can use the option in the following ways:

- You can specify a quoted *label* up to 16 characters.
- You can specify one of the keywords listed in the following table. Each keyword requests a label of the form *symbol=value*, where *symbol* is the symbol

given in the table, and *value* is the value of the central line. If the central line is not constant, only the symbol is displayed.

Keyword	Symbol Printed on Charts Produced by	
	Graphics Devices	Line Printers
S	S	S
SBAR	$\overline{\text{S}}$	$\overline{\text{S}}$
SPM	S′	S′
S0	S_0	S0

The default keyword is SBAR. The SSYMBOL= option is available only in the SCHART and XSCHART statements.

STARBDRADIUS=*value*

specifies the radius (in horizontal percent screen units) of an imaginary circle that is the outer bound for vertices of stars requested with the STARVERTICES= option. Vertices that exceed the outer bound are truncated to this value in order to prevent gross distortion of stars due to extreme values in the data. The *value* must be greater than or equal to the value specified with the STAROUTRADIUS= option. See Figure 53.12 on page 1905 or "Displaying Auxiliary Data with Stars" on page 1948. *Graphics*

STARCIRCLES=*values*

specifies reference circles that are superimposed on the stars requested with the STARVERTICES= option. All of the circles are displayed and centered at each point plotted on the primary chart. The *value* determines the diameter of the circle as follows: a *value* of zero specifies a circle with the *inner radius*, and a value of one specifies a circle with the *outer radius*. In general, a value of h specifies a circle with a radius equal to $\textit{inradius} + h \times (\textit{outradius} - \textit{inradius})$. *Graphics*

Figure 53.11 shows four circles specified with the STARCIRCLES= option. The values 0.0 and 1.0 correspond to the *inner circle* and *outer circle* (see the entries for the STARINRADIUS= and STAROUTRADIUS= options). The value 0.5 specifies a circle with a radius of $\textit{inradius} + 0.5 \times (\textit{outradius} - \textit{inradius})$ or a circle halfway between the inner circle and the outer circle. Likewise, the value 0.25 specifies a circle one-fourth of the way from the inner circle to the outer circle. Note also that the line types for the circles are specified with the LSTARCIRCLES= option. For more information, see "Displaying Auxiliary Data with Stars" on page 1948.

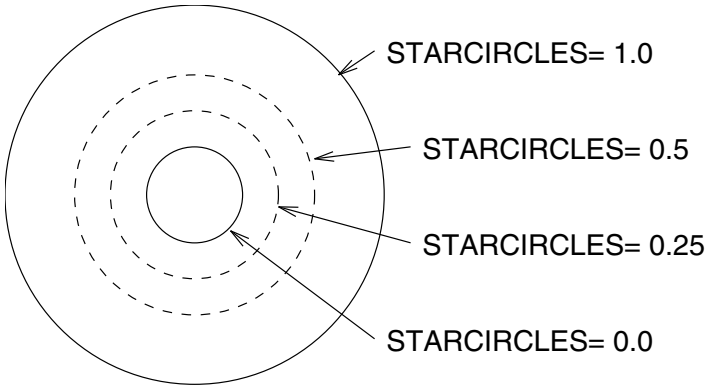


Figure 53.11. Circles Specified by STARCIRCLES=0.0 1.0 0.25 0.5

STARINRADIUS=*value*

Graphics

specifies the inner radius of stars requested with the STARVERTICES= option. The *value* must be specified in horizontal percent screen values, and it must be less than the value that is specified with the STAROUTRADIUS= option. The inner radius of a star is the distance from the center of the star to the circle that represents the lower limit of the standardized vertex variables. The lower limit can correspond to the minimum value, a multiple of standard deviations below the mean, or a lower specification limit. The default *value* is one-third of the outer radius.

Figure 53.12 illustrates five of the star options. The STARSTART= option determines the angle between the vertical axis and the first vertex. The STARINRADIUS= and STAROUTRADIUS= options specify the radii (in horizontal percent screen units) of the inner and outer circles that are associated with each star. Extremely large vertex values are truncated at the imaginary circle whose radius is specified by the STARBDRADIUS= option. The STARTYPE=RADIAL option specifies that the vertices are to be displayed as endpoints of line segments connecting each vertex to the center point. For more information, see the entries for these options or "Displaying Auxiliary Data with Stars" on page 1948.

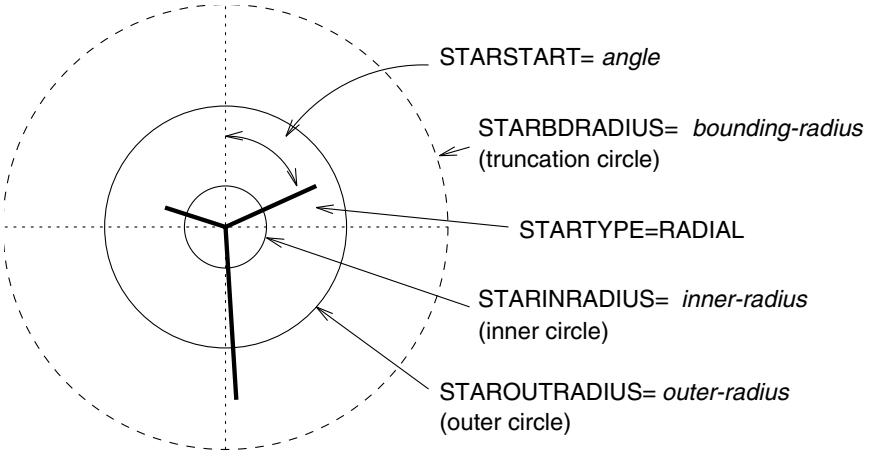


Figure 53.12. Illustration of Star Options

STARLABEL=ALL | FIRST | HIGH | LOW | OUT

specifies a method for labeling the vertices of stars requested with the STARVERTICES= option. The following table describes the method corresponding to each keyword: *Graphics*

Keyword	Method for Labeling Star Vertices
ALL	labels all vertices of all stars
FIRST	labels all vertices of the leftmost star
HIGH	labels only vertices that lie outside the outer circle
LOW	labels only vertices that lie inside the inner circle
OUT	labels only vertices that lie inside the inner circle or outside the outer circle

The label used for a particular vertex is the value of the variable _LABEL_ in the STARSPECS= data set. If this data set is not specified, or if the _LABEL_ variable is not provided, then the name of the vertex variable is used as the label. See "Displaying Auxiliary Data with Stars" on page 1948. By default, vertices are not labeled.

STARLEGEND=CLOCK | CLOCK0 | DEGREES | NONE

specifies the style of the legend used to identify the vertices of stars requested with the STARVERTICES= option. The following table describes the method corresponding to each keyword: *Graphics*

Keyword	Star Vertices Legend Style
CLOCK	identifies the vertex variables by their positions on the clock (starting with 12:00)
CLOCK0	identifies the vertex variables by their positions on the clock (starting with 0:00)
DEGREES	identifies the vertex variables by angles in degrees, with 0 degrees corresponding to 12 o'clock
NONE	suppresses the legend

See "Displaying Auxiliary Data with Stars" on page 1948. The default keyword is CLOCK.

STARLEGENDLAB=*'label'*

Graphics

specifies the label displayed to the left of the legend for stars requested with the STARLEGEND= option. The label can be up to 16 characters and must be enclosed in quotes. See "Displaying Auxiliary Data with Stars" on page 1948. The default label is *Vertices:*.

STAROUTRADIUS=*value*

Graphics

specifies the outer radius of stars requested with the STARVERTICES= option. The *value* must be specified in horizontal percent screen values. The outer radius of a star is the distance from the center of the star to the circle that represents the upper limit of the standardized vertex variables. The upper limit can correspond to the maximum value, a multiple of standard deviations above the mean, or an upper specification limit.

See Figure 53.12 on page 1905 "Displaying Auxiliary Data with Stars" on page 1948. For an example, see Figure 56.32 on page 2037. The default *value* depends on the number of subgroup positions per panel, and it is as large as possible without causing overlap of adjacent stars.

STARSPECS=*value*|*SAS-data-set*
STARSPEC=*value*|*SAS-data-set*

Graphics

specifies the method used to standardize the star vertex variables listed with the STARVERTICES= option. The method determines how the value of a vertex variable is transformed to determine the distance between the center of the star and the vertex. The STARSPECS= option also determines how the inner and outer radii of the star are to be interpreted.

A *value* of zero specifies standardization by the range of the variable. In this case, the distance between the center and the vertex is proportional to the difference between the variable value and the minimum variable value (taken across all subgroups). The inner radius of the star corresponds to the minimum variable value, and the outer radius of the star corresponds to the maximum variable value.

A positive STARSPECS= value requests standardization by a multiple of standard deviations above and below the mean. For example, STARSPECS=3 specifies that the inner radius of the star corresponds to three standard deviations below the mean,

and the outer radius corresponds to three standard deviations above the mean. Thus, a vertex variable value exactly equal to the mean is represented by a vertex whose distance to the center of the star is halfway between the inner and outer radii.

You can request a distinct method of standardization for each vertex variable by specifying a STARSPECS= *data set*. Each observation provides standardization and related information for a distinct vertex variable. The variables read from a STARSPECS= *data set* are described in the following table:

Variable	Description
CSPOKE	color of spokes used with STARTYPE=RADIAL and STARTYPE=SPOKE; this must be a character variable of length 8 or less
LABEL	label for identifying the vertex when you specify STARLEGEND=FIRST or STARLEGEND=ALL; this must be a character variable of up to 16 characters
LSL	lower specification limit
LSPOKE	line style for spokes used with STARTYPE=RADIAL, STARTYPE=SPOKE, and STARTYPE=WEDGE
NOMVAL	nominal value substituted for missing values
SIGMAS	multiple of standard deviations above and below the average
UBOUND	upper bound for truncating extremely high values
USL	upper specification limit
VAR	name of vertex variable; this must be a character variable of length 32 or less

Only the variable _VAR_ is mandatory. If you provide the variables _LSL_ and _USL_, standardization is based on the specification limits; in this case, the variable _LSL_ corresponds to the inner radius of the star, and the variable _USL_ corresponds to the outer radius of the star. If you do not provide the variables _LSL_ and _USL_, standardization is based on the value of the variable _SIGMAS_, and if you do not provide the variable _SIGMAS_, standardization is based on the range.

See "Displaying Auxiliary Data with Stars" on page 1948. If you do not specify the STARSPECS= option, each vertex variable is standardized by its range across subgroups. In other words, the minimum corresponds to the inner radius, and the maximum corresponds to the outer radius.

STARSTART=*value*

Graphics

specifies the vertex angle for the first variable in the STARVERTICES= list. Vertex angles for the remaining variables are uniformly spaced clockwise and assigned in the order listed. You can specify the *value* in the following ways:

- *Clock position:* If you specify the value as a time literal (between '0:00'T and '12:00'T), the corresponding clock position is used for the first vertex variable.

- *Degrees:* If you specify the value as a nonpositive number, the absolute value in degrees is used for the first vertex angle. Here, 0 degrees corresponds to 12:00.

The default *value* is zero, so the first vertex variable is positioned at 12:00. See Figure 53.12 on page 1905 or "Displaying Auxiliary Data with Stars" on page 1948.

STARTYPE=CORONA | POLYGON | RADIAL | SPOKE | WEDGE

Graphics

specifies the style of the stars requested with the STARVERTICES= option. The following table describes the method corresponding to each keyword.

Keyword	Star Style
CORONA	polygon with star-vertices emanating from the inner circle
POLYGON	closed polygon
RADIAL	rays emanating from the center
SPOKE	rays emanating from the inner circle
WEDGE	closed polygon with rays from the center to each vertex

See Figure 53.12 on page 1905 or "Displaying Auxiliary Data with Stars" on page 1948. "Adding Reference Circles to Stars" on page 1950 describes the inner and outer circles, and "Specifying the Style of Stars" on page 1952 provides examples of each value of the STARTYPE= option. The default keyword is POLYGON.

STARVERTICES=*variable* **|** **(***variable-list***)**

Graphics

superimposes a star (polygon) at each point on the primary chart. The star is centered at the point, and the distance between the center and each star vertex represents the standardized value of a *variable* in the STARVERTICES= list. The *variables* must be provided in the input data set.

The star display is suggested as a method for monitoring quantitative variables (such as environmental factors) that are measured simultaneously with the process variable. For examples and details, see "Displaying Auxiliary Data with Stars" on page 1948. By default, stars are not superimposed on the chart.

STDDEVIATIONS
STDDEVS

specifies that the estimate of the process standard deviation σ is to be calculated from subgroup standard deviations. This, in turn, affects the calculation of control limits; for details, see "Methods for Estimating the Process Standard Deviation" in the chapter for the chart statement in which you are interested. By default, the estimate of σ is calculated from subgroup ranges, except with the BOXCHART statement, where subgroup standard deviations are used by default.

If you specify the STDDEVIATIONS option and read summary data from a HISTORY= data set, the data set must contain a subgroup standard deviation variable for each *process*. Conversely, if you omit the STDDEVIATIONS option, the HISTORY= data set must contain a subgroup range variable for each *process* listed in the chart statement.

You should specify STDDEVIATIONS when your subgroup sample sizes are large (typically, 15 or greater). The STDDEVIATIONS option is available only in the MCHART and XCHART statements.

SUBGROUPN=*value*
SUBGROUPN=*variable*

specifies the subgroup sample sizes as a constant *value* or as the values of a *variable* in the DATA= data set. The SUBGROUPN= option is available only in the CCHART, NPCHART, PCHART, and UCHART statements.

You must specify SUBGROUPN= in the NPCHART, PCHART, and UCHART statements when your input data set is a DATA= data set. If you are using a CCHART statement, the SUBGROUPN= option is available only when your input data set is a DATA= data set. For the CCHART statement, the default value of the SUBGROUPN= option is one.

If you specify multiple *processes* in a chart statement, the SUBGROUPN= option is used with all of the *processes* listed.

SYMBOLCHARS='*character-list'*

specifies a list of characters used to mark the points plotted on charts produced with a line printer when a *symbol-variable* is used. See "Displaying Stratification in Levels of a Classification Variable" on page 1931. *Line Printer*

Each character is associated with a level (unique value) of the *symbol-variable* and is used to mark points associated with that value. For example, consider the following statements:

```
proc shewhart;
   xrchart gap*shift=machine / symbolchars='12345';
run;
```

Here the *symbol-variable* is MACHINE. The $\bar{X}$ and R charts use a '1' to mark points associated with the first unique value of MACHINE, a '2' to mark points associated with the second unique value of MACHINE, and so on.

If the number of levels of the *symbol-variable* exceeds the number of *characters*, the last character listed is used for points associated with the additional values. Thus, in the preceding example, if there are six levels of MACHINE, points with the fifth and six values are indicated by '5'.

The default *character-list* is ABCDEFGHIJKLMNOPQRSTUVWXYZ*. Thus, the procedure uses 'A' for the first unique value of the *symbol-variable*, 'B' for the second unique value, and so on. An asterisk is used for points associated with the 27^{th} and subsequent levels when the *symbol-variable* has more than 26 levels.

SYMBOLLEGEND=LEGEND*n*
SYMBOLLEGEND=NONE

controls the legend for the levels of a *symbol-variable* (see "Displaying Stratification in Levels of a Classification Variable" on page 1931). You can specify *Graphics*

SYMBOLLEGEND=LEGEND*n*, where *n* is the number of a LEGEND statement defined previously. You can specify SYMBOLLEGEND=NONE to suppress the default legend.

SYMBOLORDER=DATA | INTERNAL | FORMATTED
SYMORD=DATA | INTERNAL | FORMATTED

specifies the order in which symbols are assigned for levels of *symbol-variable*. The DATA keyword assigns symbols to values in the order in which values appear in the input data. This is how symbols were assigned in Release 6.12 and earlier releases of SAS/QC software. The INTERNAL keyword assigns symbols based on sorted order of internal values of *symbol-variable* and FORMATTED assigns them based on sorted formatted values. The default value is FORMATTED.

TABLE <(EXCEPTIONS)>
TABLES <(EXCEPTIONS)>

creates a basic table of the subgroup values, the subgroup sample sizes, the subgroup summary statistics, and the upper and lower control limits. Rows of the table correspond to subgroups. The keyword EXCEPTIONS (enclosed in parentheses) is optional and restricts the tabulation to those subgroups for which the control limits are exceeded or a test for special causes is positive.

You can request extended versions of the basic table by specifying one or more of the following options: TABLEBOX, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS. Specifying the TABLEALL option is equivalent to specifying all of these options, and it provides the most extensive table.

TABLEALL <(EXCEPTIONS)>

tabulates the information on the control chart and is equivalent to specifying all of the following options: TABLES, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUT, and TABLETESTS. If you specify the TABLEALL option in a BOXCHART statement, the TABLEBOX option is also implied. The keyword EXCEPTIONS (enclosed in parentheses) is optional and restricts the tabulation to those subgroups for which the control limits are exceeded or a test for special causes is positive. You can use the OUTTABLE= option to create a data set that saves the information tabulated with the TABLEALL option.

TABLEBOX <(EXCEPTIONS)>

augments the basic table created by the TABLES option with columns for the minimum, 25th percentile, median, 75th percentile, and maximum of the observations in a subgroup. The TABLEBOX option is available only in the BOXCHART statement. The keyword EXCEPTIONS (enclosed in parentheses) is optional and restricts the tabulation to those subgroups for which the control limits are exceeded or a test for special causes is positive.

TABLECENTRAL <(EXCEPTIONS)>
TABLEC <(EXCEPTIONS)>

augments the basic table created by the TABLES option with columns for the values of the central lines. The keyword EXCEPTIONS (enclosed in parentheses) is optional

and restricts the tabulation to those subgroups for which the control limits are exceeded or a test for special causes is positive.

TABLEID <**(EXCEPTIONS)**>
augments the basic table created by the TABLES option with a column for each of the ID variables. The keyword EXCEPTIONS (enclosed in parentheses) is optional and restricts the tabulation to those subgroups for which the control limits are exceeded or a test for special causes is positive.

TABLELEGEND <**(EXCEPTIONS)**>
TABLELEG <**(EXCEPTIONS)**>
adds a legend to the basic table created by the TABLES option. The legend describes the tests for special causes that were requested with the TESTS= option and for which a positive signal is found for at least one subgroup. The keyword EXCEPTIONS (enclosed in parentheses) is optional and restricts the tabulation to those subgroups for which the control limits are exceeded or a test for special causes is positive.

TABLEOUTLIM <**(EXCEPTIONS)**>
TABLEOUT <**(EXCEPTIONS)**>
augments the basic table created by the TABLES option with columns indicating which control limits (if any) are exceeded. The keyword EXCEPTIONS (enclosed in parentheses) is optional and restricts the tabulation to those subgroups for which the control limits are exceeded or a test for special causes is positive.

TABLETESTS <**(EXCEPTIONS)**>
augments the basic table created by the TABLES option with a column that indicates which of the tests for special causes (requested with the TESTS= option) are positive. The column contains the numbers of all the tests that are positive at a particular subgroup. The keyword EXCEPTIONS (enclosed in parentheses) is optional and restricts the tabulation to those subgroups for which the control limits are exceeded or a test for special causes is positive.

TARGET=*value-list*
provides target values used to compute the capability index C_{pm}, which is saved in the OUTLIMITS= data set. If you provide more than one *value*, the number of *values* must match the number of *processes* listed in the chart statement. If you specify only one *value*, it is used for all the *processes*.

CAUTION: You can use the TARGET= options only in conjunction with the LSL= and USL= options. For more information, see "Capability Indices" on page 1774 and "Output Data Sets" in the chapter for the chart statement in which you are interested. Also see the entries for the LSL= and USL= options. The TARGET= option is available in the BOXCHART, IRCHART, MCHART, MRCHART, RCHART, SCHART, XCHART, XRCHART, and XSCHART statements.

TEST2RESET=*variable*
TEST2RESET=*value*
enables tests for special causes to be reset in a secondary chart. The specified variable must be a character variable of length 8, or length 16 if customized tests are requested. The variable values have the same format as those of the _TESTS_ variable in a

TABLE= data set. A test that is flagged by the TEST2RESET= value for a given subgroup is reset starting with that subgroup. That means a positive result for the test can include the given subgroup only if it is the first subgroup in the pattern. For example, the value "12345678" for the TEST2RESET= variable will reset all standard tests for special causes.

TEST2RUN=*run-length*

specifies the length of the pattern for Test 2 requested with the TESTS= and TESTS2= options. The values allowed for the *run-length* are 7, 8, 9, 11, 14, and 20. The form of the test for each *run-length* value is given in the following table. The default *run-length* is 9. See Chapter 55, "Tests for Special Causes," on page 1975 for more information.

Run-length	Number of Points on One Side of the Central Line
7	7 in a row
8	8 in a row
9	9 in a row
11	at least 10 out of 11 in a row
14	at least 12 out of 14 in a row
20	at least 16 out of 20 in a row

TEST3RUN=*run-length*

specifies the length of the pattern for Test 3 requested with the TESTS= and TESTS2= options. Test 3 searches for a pattern of steadily increasing or decreasing values, where the length of the pattern is at least the value given as the *run-length*. The values allowed for the *run-length* are 6, 7, and 8. The default *run-length* is 6. See Chapter 55, "Tests for Special Causes," on page 1975 for more information.

TESTACROSS

specifies that tests for special causes requested with the TESTS= or TESTS2= options are to be applied without regard to phases (blocks of consecutive subgroups) determined by the READPHASES= option and the variable _PHASE_ in the input data set. If you specify the READPHASES= option but do not specify the TESTACROSS option, tests for special causes are applied within (but not across) phases. See Chapter 55, "Tests for Special Causes," on page 1975.

TESTCHAR='*character***'**

Line Printer

specifies the character for the line segments that connect any sequence of points for which a test for special causes (requested with the TESTS= or TESTS2= option) is positive. The default *character* is the number of the test (with values 1 to 8).

TESTFONT=*font*
LABELFONT=*font*

Graphics

specifies a software font for labels requested with the ALLLABEL=, ALLLABEL2=, OUTLABEL=, OUTLABEL2=, STARLABEL=, TESTLABEL=, and TESTLABEL*n*= options. Hardware characters are used by default.

TESTHEIGHT=*value*
LABELHEIGHT=*value*

specifies the height (in vertical percent screen units) for labels requested with the ALLLABEL=, ALLLABEL2=, OUTLABEL=, OUTLABEL2=, STARLABEL=, TESTLABEL=, and TESTLABELn= options. The default height is the height specified with the HEIGHT= option or the HTEXT= option in the GOPTIONS statement. *Graphics*

TESTLABBOX

requests that labels for subgroups with positive tests for special causes are positioned so they do not overlap. The labels are enclosed in boxes that are connected to the associated subgroup points with line segments. *Graphics*

TESTLABEL='*label***'**
TESTLABEL=(*variable***)**
TESTLABEL=TESTINDEX
TESTLABEL=SPACE
TESTLABEL=NONE

provides labels for points at which one of the tests for special causes (requested with the TESTS= or TESTS2= option) is positive. The values for the TESTLABEL= option are as follows:

- You can specify a *label* of up to 16 characters enclosed in quotes. This label is displayed at all points where a test is signaled.
- You can specify a *variable* (enclosed in parentheses) whose values are used as labels. The *variable* must be provided in the input data set, and it can be numeric or character. If the *variable* is character, its length cannot exceed 16. For each subgroup of observations at which a test is signaled, the formatted value of the *variable* in the observations is used to label the point representing the subgroup. If you are reading a DATA= data set with multiple observations per subgroup, the values of the *variable* should be identical for observations within a subgroup.
- You can specify TESTINDEX to label points with the single-digit *index* that requested the test in a TESTS= or TESTS2= list. If the test was requested with a customized *pattern* in a TESTS= or TESTS2= list, then points are labeled with the letter that you specified using the CODE= option.
- You can specify SPACE to request a label of the form *Test k*. This is slightly more legible than the default label of the form *Testk* (a description of *Testk* follows).
- You can specify NONE to suppress labeling.

If you do not use the TESTLABEL= option, the default label is of the form *Testk*, where k is the index of the test as requested with the TESTS= or TESTS2= options, or k is the CODE= *character* of the test as requested in a pattern specified with the TESTS= or TESTS2= options.

See Chapter 55, "Tests for Special Causes," on page 1975. Related options include OUTLABEL=, OUTLABEL2=, TESTFONT=, TESTHEIGHT=, and TESTLABELn=.

TESTLABEL*n*='*label*'

specifies a *label* for points at which the test for special causes requested with the *index* n in a TESTS= or TESTS2= list is positive. The *index* n can be a number from 1 to 8. The TESTLABELn= option overrides a TESTLABEL= option and the default label *Test n*. The *label* that you specify with the TESTLABELn= option can be up to 16 characters and must be enclosed in quotes.

See Chapter 55, "Tests for Special Causes," on page 1975. Related options are TESTFONT=, TESTHEIGHT=, and TESTLABEL=.

TESTNMETHOD=STANDARDIZE

applies the tests for special causes requested with the TESTS= and TESTS2= options to standardized test statistics when the subgroup sample sizes are not constant. This method was suggested by Nelson (1994). See Chapter 55, "Tests for Special Causes," on page 1975. By default, the tests are not applied to data with varying subgroup sample sizes.

TESTOVERLAP

applies tests for special causes (requested with the TESTS= or TESTS2= option) to overlapping patterns of points.

The TESTOVERLAP option modifies the way in which the search for a subsequent pattern is done when a pattern is encountered. If you omit the TESTOVERLAP option, the search begins with the first subgroup after the current pattern ends. If you specify the TESTOVERLAP option, the search begins with the second subgroup in the current pattern.

The following statements illustrate the use of the TESTOVERLAP option:

```
proc shewhart;
   xrchart width*hour / test=3;
   xrchart width*hour / test=3 testoverlap;
run;
```

Test 3 looks for six subgroup means in a row steadily increasing or decreasing. Suppose that the subgroup means of WIDTH are steadily increasing for HOUR=5, 6, 7, 8, 9, 10, and 11. The first XRCHART statement will signal that Test 3 is positive at HOUR=10 but not at HOUR=11, since the search for the next pattern begins with HOUR=11. The second XRCHART statement will signal that Test 3 is positive at HOUR=10 and HOUR=11, since the search for the next pattern begins with HOUR=6 and thus finds a second pattern ending with HOUR=11. See Chapter 55, "Tests for Special Causes," on page 1975 for more information.

CAUTION: Specifying TESTOVERLAP affects the interpretation of the standard tests for special causes, because a particular point can contribute to more than one positive test. Typically, this option should not be used.

TESTRESET=*variable*
TESTRESET=*value*

enables tests for special causes to be reset in a primary chart. The specified variable must be a character variable of length 8, or length 16 if customized tests are requested. The variable values have the same format as those of the _TESTS_ variable in a TABLE= data set. A test that is flagged by the TESTRESET= value for a given subgroup is reset starting with that subgroup. That means that a positive result for the test can include the given subgroup only if it is the first subgroup in the pattern. For example, the value "12345678" for the TESTRESET= variable will reset all standard tests for special causes.

TESTS=*index-list*
TESTS=*customized-pattern-list*

requests one or more tests for special causes, which are also known as *runs tests*, *pattern tests*, and *Western Electric rules*. These tests detect particular nonrandom patterns in the points plotted on the primary control chart. The occurrence of a pattern, referred to as a *signal*, suggests the presence of a special cause of variation.

Each pattern is defined in terms of zones A, B, and C, which are constructed by dividing the interval between the control limits into six equally spaced subintervals. Zone A is the union of the subintervals immediately below the upper control limit and immediately above the lower control limit. Zone C is the union of the subintervals immediately above and below the central line. Zone B is the union of the subintervals between zones A and C. See Figure 55.1 on page 1978 for an illustration of test zones.

You can use the TESTS= option in three ways. First, you can specify an *index-list* to request a combination of standard tests (this is the approach most commonly used). Second, you can specify a *customized-pattern-list* to request a combination of tests based on customized patterns. Third, you can specify a list consisting of both *indexes* and *customized-patterns*. The first two approaches are described as follows.

Standard tests. The following table lists the standard tests that you can request by specifying TEST=*index-list*. The tests are indexed according to the sequence used by Nelson (1984, 1985).

You can specify any combination of the eight *indexes* with an explicit list or with an implicit list, as in the following example:

```
proc shewhart;
   xrchart width*hour / tests=1 2 3 4;
   xrchart width*hour / tests=1 to 4;
run;
```

The TESTS= option is available in all but the RCHART and SCHART statements. Use only tests 1, 2, 3, and 4 in the CCHART, NPCHART, PCHART, and UCHART statements. By default, the TESTS= option is not applied in any chart statement unless the control limits are 3σ limits. You can use the NO3SIGMACHECK option to request tests for special causes when you use the SIGMAS= option to specify control limits other than 3σ limits.

Index	Pattern Description
1	one point beyond Zone A (outside the control limits)
2	nine points in a row in Zone C or beyond on one side of the central line (see the entry for the TEST2RUN option)
3	six points in a row steadily increasing (see the entry for the TEST3RUN option)
4	fourteen points in a row alternating up and down
5	two out of three points in a row in Zone A or beyond
6	four out of five points in a row in Zone B or beyond
7	fifteen points in a row in Zone C on either or both sides of the central line
8	eight points in a row on either or both sides of the central line with no points in Zone C

Customized tests. Although the standard tests supported by the TESTS= option are appropriate for the vast majority of control chart applications, there may be situations in which you want to work with customized tests. You can define your own tests by specifying TESTS=*customized-pattern-list*. There are two types of patterns that you can include in this list: *T-patterns* and *M-patterns*.

Use a T-pattern to request a search for k out of m points in a row in the interval (a, b). The required syntax for a T-pattern is

T(K=*k* **M=***m* **LOWER=***a* **UPPER=***b* **CODE=***character* **LABEL=***'label'***)**

The default value for SCHEME= is ONESIDED. The options for a T-pattern are summarized in the following table:

Option	Description
K=*k*	number of points
M=*m*	number of consecutive points
LOWER=*value*	lower limit of interval (a, b)
UPPER=*value*	upper limit of interval (a, b)
SCHEME=ONESIDED	one-sided scheme using (a, b)
SCHEME=TWOSIDED	two-sided scheme using $(a, b) \cup (-b, -a)$
CODE=*character*	identifier for test (A-H)
LABEL=*'label'*	label for points if signal
LEGEND=*'legend'*	legend used with the TABLELEGEND option

Use an M-pattern to request a search for k points in a row increasing or decreasing. The required syntax for an M-pattern is

M(K=*k* **DIR=***direction* **CODE=***character* **LABEL=***'label'***)**

The options for an M-pattern are summarized in the following table:

Option	Description
K=*k*	number of points
DIR=INC	increasing pattern
DIR=DEC	decreasing pattern
CODE=*character*	identifier for test (A-H)
LABEL='*label*'	label for points if signal
LEGEND='*legend*'	legend used with the TABLELEGEND option

For details on the TESTS= option, see Chapter 55, "Tests for Special Causes," on page 1975. Related options include CTEST=, CZONES=, LTEST=, TABLETESTS, TABLELEGEND, TEST2RUN=, TEST3RUN=, TESTACROSS, TESTCHAR=, TESTLABEL=, TESTLABELn=, TESTNMETHOD=, TESTOVERLAP, TESTS2=, ZONES, ZONECHAR=, and ZONELABELS.

TESTS2=*index-list*
TESTS2=*customized-pattern-list*

requests one or more tests for special causes for an R chart or s chart. The syntax for the TESTS2= option is identical to the syntax for the TESTS= option. The TESTS2= option is available in the MRCHART, RCHART, SCHART, XRCHART, and XSCHART statements. For details on the TESTS2= option, see Chapter 55, "Tests for Special Causes," on page 1975. Related options include CTEST=, CZONES=, LTEST=, TABLETESTS, TABLELEGEND, TEST2RUN=, TEST3RUN=, TESTACROSS, TESTCHAR=, TESTLABEL=, TESTLABELn=, TESTNMETHOD=, TESTOVERLAP, TESTS=, ZONES2, ZONECHAR=, and ZONE2LABELS.

TESTSYMBOL=*symbol*
TESTSYM=*symbol*

specifies the symbol for plotting subgroups with positive tests for special causes. *Graphics*

TESTSYMBOLHT=*value*
TESTSYMHT=*value*

specifies the height of the symbol used to plot subgroups with positive tests for special causes. *Graphics*

TOTPANELS=*n*

specifies the total number of panels to be used to display the chart. This option overrides the NPANEL= option.

TRENDVAR=*variable* **| (***variable-list***)**

specifies a list of trend variables, one for each *process* listed in the chart statement. The TRENDVAR= option is available only in the BOXCHART, MCHART, and XCHART statements and only when your input data set is a DATA= or HISTORY= data set.

The values of the trend variables are subtracted from the values of the corresponding process variables (if you read a DATA= data set) or subgroup mean variables (if you read a HISTORY= data set). The chart is then created for the residuals (differences),

and the trend values are plotted in a secondary chart. If you specify a single trend variable and two or more *processes*, the trend variable is used with each *process*.

The TRENDVAR= option does not apply if you are reading a TABLE= data set. In this case, the procedure produces a trend chart only if the variable _TREND_ is included in the TABLE= data set.

For more details, see "Displaying Trends in Process Data" on page 1957. Related options include NOTRENDCONNECT, SEPARATE, SPLIT=, WTREND=, and YPCT1=.

TURNALL
TURNOUT

turns the labels produced by the ALLLABEL=, ALLLABEL2=, OUTLABEL= and OUTLABEL2= options so that they are strung out vertically. By default, labels are arranged horizontally.

TURNHLABELS
TURNHLABEL

turns the major tick mark labels for the horizontal (subgroup) axis so that they are strung out vertically. By default, labels are arranged horizontally.

If you are using a graphics device, you should specify a software font (with the FONT= option) in conjunction with the TURNHLABELS option. Otherwise, the labels may be displayed with a mixture of hardware and software fonts.

Note: Turning the labels vertically may leave insufficient room on the screen or page for a chart.

TYPE=*value*

specifies the *value* of the _TYPE_ variable in the OUTLIMITS= data set, which in turn indicates whether certain parameter variables in this data set represent estimates or standard (known) values.

If you are using a chart statement that creates a variables chart, _TYPE_ is a bookkeeping variable that indicates whether the values of the variables _MEAN_ and _STDDEV_ in the OUTLIMITS= data set are estimates or standard values of the process mean μ and standard deviation σ. The following table summarizes the *values* that you can specify:

Value	_MEAN_	_STDDEV_
ESTIMATE	estimate	estimate
STDMU	standard	estimate
STDSIGMA	estimate	standard
STANDARD	standard	standard

The default *value* is ESTIMATE, unless you specify standard values for μ or σ with the MU0= or SIGMA0= options.

For PCHART and NPCHART statements, the *value* you specify for the TYPE= option can be either ESTIMATE or STANDARD, indicating that the value of the vari-

able _P_ in the OUTLIMITS= data set is an estimate or standard value of the proportion p of nonconforming items. The default *value* is ESTIMATE, unless you specify a standard value for p with the P0= option.

For UCHART and CCHART statements, the *value* you specify for the TYPE= option can be either ESTIMATE or STANDARD, indicating that the value of the variable _U_ in the OUTLIMITS= data set is an estimate or standard value of the average number u of nonconformities per unit. The default *value* is ESTIMATE, unless you specify a standard value for u with the U0= option.

U0=*value*

specifies a known (standard) value u_0 for the average number u of nonconformities per unit produced by the process. By default, u_0 is estimated from the data. The U0= option is available only in the CCHART and UCHART statements.

Note: As an alternative to specifying the U0= option, you can read a predetermined value for u_0 from the variable _U_ in a LIMITS= data set. For details, see "Input Data Sets" in the chapter for the chart statement in which you are interested.

UCLLABEL='*label***'**

specifies a label for the upper control limit in the primary chart. The label can be up to 16 characters. Enclose the label in quotes. The default label is of the form *UCL=value* if the control limit has a fixed value; otherwise, the default label is *UCL*. Related options are UCLLABEL2=, LCLLABEL=, and LCLLABEL2=.

UCLLABEL2='*label***'**

specifies a label for the upper control limit in the secondary chart. The label can be up to 16 characters. Enclose the label in quotes. The default label is of the form *UCL=value* if the control limit has a fixed value; otherwise, the default label is *UCL*. This option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements. Related options are LCLLABEL2=, LCLLABEL=, and UCLLABEL=.

USL=*value-list*

provides upper specification limits used to compute capability indices. If you provide more than one *value*, the number of *values* must match the number of *processes* listed in the chart statement. If you specify only one *value*, it is used for all the *processes*.

The SHEWHART procedure uses the specification limits to compute capability indices, and it saves the limits and indices in the OUTLIMITS= data set. For more information, see "Capability Indices" on page 1774 and "Output Data Sets" in the chapter for the chart statement in which you are interested. A related option is LSL=. The USL= option is available in the BOXCHART, IRCHART, MCHART, MRCHART, RCHART, SCHART, XCHART, XRCHART, and XSCHART statements.

USYMBOL='*label***'**
USYMBOL=U | UBAR | UPM | UPM2 | U0

specifies a label for the central line in a u chart. You can use the option in the following ways:

- You can specify a quoted *label* up to 16 characters.

- You can specify one of the keywords listed in the following table. Each keyword requests a label of the form *symbol=value*, where *symbol* is the symbol given in the table, and *value* is the value of the central line. If the central line is not constant, only the symbol is displayed.

Keyword	Symbol Printed on Charts Produced by	
	Graphics Devices	Line Printers
U	U	U
UBAR	$\overline{\mathrm{U}}$	$\overline{\mathrm{U}}$
UPM	U'	U'
UPM2	U''	U''
U0	U_0	U0

The default keyword is UBAR. The USL= option is available only in the UCHART statement.

VAXIS=*value-list*
VAXIS=AXIS*n*

specifies major tick mark values for the vertical axis of a primary chart. The *values* must be listed in increasing order, must be evenly spaced, and must span the range of summary statistics and control limits displayed in the chart. You can specify the *values* with an explicit list or with an implicit list, as shown in the following example:

```
proc shewhart;
   xrchart width*hour / vaxis=0 2 4 6 8;
   xrchart width*hour / vaxis=0 to 8 by 2;
run;
```

If you are using a graphics device, you can also specify a previously defined AXIS statement with the VAXIS= option. Related options are HAXIS= and VAXIS2=.

VAXIS2=*value-list*
VAXIS2=AXIS*n*

specifies major tick mark values for the vertical axis of a secondary chart. The specifications and restrictions are the same as for the VAXIS= option. The VAXIS2= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option. Related options are HAXIS= and VAXIS=.

VFORMAT=*format*

specifies a format to be used for displaying tick mark labels on the vertical axis of a primary chart.

VFORMAT2=*format*

specifies a format to be used for displaying tick mark labels on the vertical axis of a secondary chart.

VMINOR=*n*
VM=*n*

specifies the number of minor tick marks between each major tick mark on the vertical axis. No values are printed on the minor tick marks. By default, VMINOR=0. *Graphics*

VOFFSET=*value*

specifies the length in percent screen units of the offset at the ends of the vertical axis. *Graphics*

VREF=*value-list*
VREF=*SAS-data-set*

draws reference lines perpendicular to the vertical axis on the primary chart. Reference line values can be expressed as simple values or as multiples of the standard error of the subgroup summary statistic. You can use this option in the following ways:

- Specify the *values* for the lines with a VREF= list. Examples of the VREF= option follow:

```
vref=20
vref=20 40 80
vref=(2.5 sigma)
vref=20 (1.5 2.0 2.5 sigma) 80
```

  Values expressed as multiples of σ must be enclosed in parentheses with the SIGMA keyword.

- Specify the values for the lines as the values of a numeric variable named _REF_ in a VREF= data set. Optionally, you can provide labels for the lines as values of a variable named _REFLAB_, which must be a character variable of length 16 or less. If you want distinct reference lines to be displayed in charts for different *processes* specified in the chart statement, you must include a character variable of length 32 or less named _VAR_, whose values are the *processes*. If you do not include the variable _VAR_, all of the lines are displayed in all of the charts. If you want to display reference lines whose values are multiples of σ, you must include a character variable named _TYPE_, whose values are "VALUES" or "SIGMAS." The value of _TYPE_ indicates whether the reference line value is expressed as a simple value or as a multiple of σ.

  Each observation in the VREF= data set corresponds to a reference line. If BY variables are used in the input data set (DATA=, HISTORY=, or TABLE=), the same BY variable structure must be used in the VREF= data set unless you specify the NOBYREF option.

This option can be used to add warning limits to be displayed on a chart.

Related options are CVREF=, LVREF=, NOBYREF, VREFCHAR=, VREFLABELS=, and VREFLABPOS=.

VREF2=*value-list*
VREF2=*SAS-data-set*
draws reference lines perpendicular to the vertical axis on the secondary chart. The conventions for specifying the VREF2= option are identical to those for specifying the VREF= option. Related options are CVREF=, LVREF=, NOBYREF, VREFCHAR=, VREF2LABELS=, and VREFLABPOS=. The VREF2= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option.

VREF2LABELS='*label1*' . . . '*labeln*'
VREF2LAB='*label1*' . . . '*labeln*'
specifies labels for the reference lines requested by the VREF2= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters. The VREF2LABELS= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option.

VREFCHAR='*character*'
Line Printer
specifies the character used to form the reference lines requested by the VREF= and VREF2= options for a line printer. The default is the hyphen (-).

VREFLABELS='*label1*' . . . '*labeln*'
specifies labels for the reference lines requested by the VREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

VREFLABPOS=*n*
specifies the horizontal position of the VREFLABEL= and VREF2LABEL= labels, as described in the following table. By default, n=1.

n	Label Position
1	left-justified in subplot area
2	right-justified in subplot area
3	left-justified in right margin

Figure 53.13 illustrates label positions for values of the VREFLABPOS= option when the VREF= and VREFLABELS= options are as follows:

```
vref       = 8 9 11
vreflabels = 'Eight' 'Nine' 'Eleven'
```

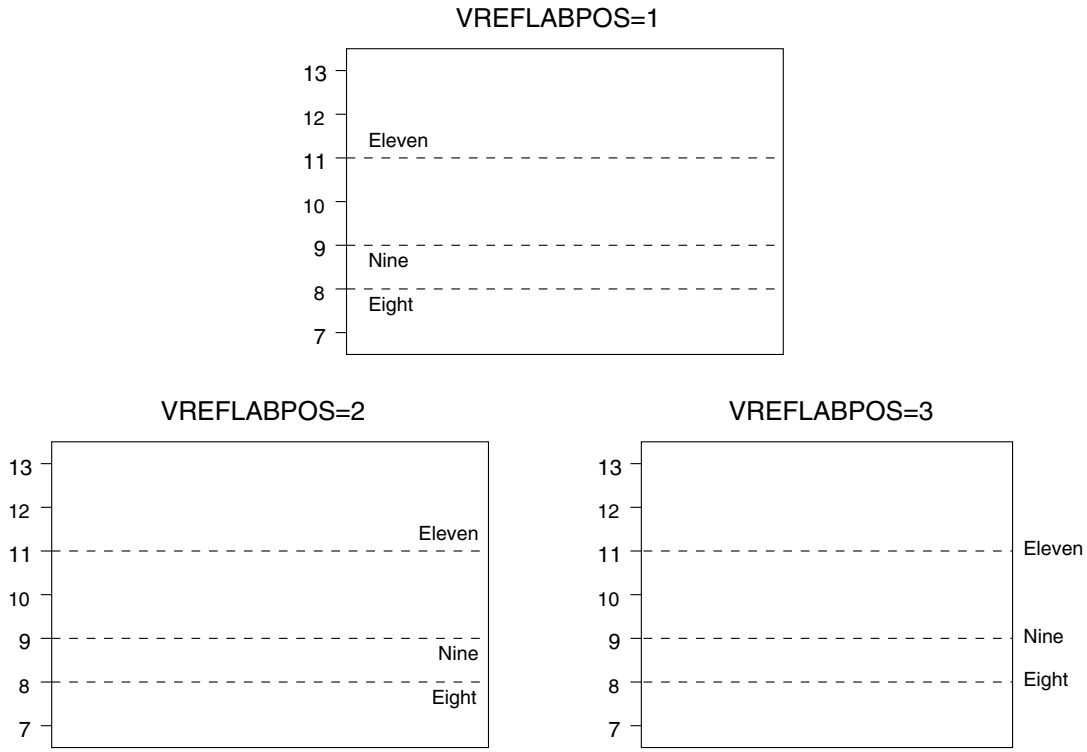


Figure 53.13. Positions for Reference Line Labels

VZERO
forces the origin to be included in the vertical axis for a primary chart.

VZERO2
forces the origin to be included in the vertical axis for a secondary chart.

WAXIS=*n*
specifies the width in pixels for the axis and frame lines. By default, $n = 1$. *Graphics*

WEBOUT=*SAS-data-set*
produces an output data set containing all the data in an OUTTABLE= data set plus graphics coordinates for points (subgroup summary statistics) that are displayed on a control chart. You can use an WEBOUT= data set to facilitate the development of web-based applications. See Chapter 57, "Interactive Control Charts," for details. *Graphics*

WGRID=*n*
specifies the width in pixels for grid lines requested with the ENDGRID and GRID options. By default, $n = 1$. *Graphics*

WLIMITS=*n*
specifies the width in pixels for the control limits and central line. By default, $n = 1$. *Graphics*

WNEEDLES=*n*

Graphics

specifies the width in pixels of needles connecting plotted points to the central line, as requested with the NEEDLES option. If you use the WNEEDLES= option, you do not need to specify the NEEDLES option. By default, $n = 1$.

WOVERLAY=(*value-list***)**

Graphics

specifies the widths in pixels for the line segments connecting points on primary chart overlay plots. Widths in the WOVERLAY= list are matched with variables in the corresponding positions in the OVERLAY= list.

WOVERLAY2=(*value-list***)**

Graphics

specifies the widths in pixels for the line segments connecting points on secondary chart overlay plots. Widths in the WOVERLAY2= list are matched with variables in the corresponding positions in the OVERLAY2= list.

WSTARCIRCLES=*n*

Graphics

specifies the width in pixels of the outline of circles requested by the STARCIRCLES= option. See "Displaying Auxiliary Data with Stars" on page 1948. By default, $n = 1$.

WSTARS=*n*

Graphics

specifies the width in pixels of the outline of stars requested by the STARVERTICES= option. See "Displaying Auxiliary Data with Stars" on page 1948. By default, $n = 1$.

WTESTS=*n*
WTEST=*n*

Graphics

specifies the width in pixels of the line segments that connect patterns of points for which a test for special causes (requested with the TESTS= or TESTS2= option) is positive. By default, $n = 1$.

WTREND=*n*

Graphics

specifies the width in pixels of the line segments that connect points on trend charts requested with the TRENDVAR= option. By default, $n = 1$. The WTREND= option is available in the BOXCHART, MCHART, and XCHART statements.

XSYMBOL='*label***'**
XSYMBOL=*keyword*

specifies a label for the central line in an $\bar{X}$ chart or a median chart. You can use the option in the following ways:

- You can specify a quoted *label* up to 16 characters.
- You can specify one of the *keywords* listed in the following table. Each *keyword* requests a label of the form *symbol*=*value*, where *symbol* is the symbol given in the table, and *value* is the value of the central line. If the central line is not constant, only the symbol is displayed.

Keyword	Symbol Printed on Charts Produced by	
	Graphics Devices	Line Printers
MBAR	$\overline{\mathrm{M}}$	$\overline{\mathrm{M}}$
MTIL	$\tilde{\mathrm{M}}$	$\tilde{\mathrm{M}}$
MU	μ	MU
MU0	μ_0	MU0
XBAR	$\overline{\mathrm{X}}$	$\overline{\mathrm{X}}$
XBAR2	$\overline{\overline{\mathrm{X}}}$	$\overline{\overline{\mathrm{X}}}$
XBARPM	$\overline{\mathrm{X}}'$	$\overline{\mathrm{X}}'$
XBAR0	$\overline{\mathrm{X}}_0$	$\overline{\mathrm{X}}0$
XBAR0PM	$\overline{\mathrm{X}}_0'$	$\overline{\mathrm{X}}0'$

For the IRCHART statement, the default *keyword* is XBAR. For the MCHART and MRCHART statements, the default *keyword* is MBAR. For all other chart statements, the default *keyword* is XBAR2. The XSYMBOL= option is available in the BOXCHART, IRCHART, MCHART, MRCHART, XCHART, XRCHART, and XSCHART statements.

YPCT1=*value*

specifies a percent (ranging from 0 to 100) that determines the length of the vertical axis for the primary chart in proportion to the sum of the lengths of the vertical axes for the primary and secondary charts. For example, you can specify YPCT1=50 in an XRCHART statement to request that the vertical axes for the $\bar{X}$ and R charts have the same length. The default *value* is 60. The YPCT1= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option.

YSCALE=PERCENT

scales the vertical axis on a p chart in percent units. The YSCALE= option is available only in the PCHART statement.

ZEROSTD

ZEROSTD=NOLIMITS

specifies that a control chart is to be constructed and displayed regardless of whether the estimated process standard deviation $\hat{\sigma}$ is zero. When $\hat{\sigma}$ is zero, the control limits are degenerate (collapsed around the central line), and the chart simply serves as a placeholder, particularly when a series of charts is to be created. Specify ZEROSTD=NOLIMITS to suppress the display of the degenerate limits. By default, a chart is not displayed when $\hat{\sigma}$ is zero.

ZONE2LABELS

adds the labels A, B, and C to zone lines requested with the ZONES2 or ZONE2VALUES options. The ZONE2LABELS option is available in the MRCHART, RCHART, SCHART, XRCHART, and XSCHART statements.

ZONE2VALUES
labels R or s chart zones lines with their values. If the ZONE2VALUES option is specified the ZONES2 option is not required.

ZONECHAR='*character***'**
Line Printer
specifies the character used to form the zone lines requested by the ZONES option. See the entry for the TESTS= option for a description of the zones. You do not need to specify the ZONES option if you specify the ZONECHAR= option. By default, the line between Zone A and Zone B uses the character 'B', and the line between Zone B and Zone C uses the character 'C'. Related options are TESTS=, TESTS2=, ZONES, and ZONES2.

ZONELABELS
adds the labels A, B, and C to zone lines requested with the ZONES or ZONEVALUES options. The ZONELABELS option is not available in the RCHART or SCHART statements.

ZONES
adds lines to a primary chart that delineate zones A, B, and C for standard tests requested with the TESTS= option. Related options are CZONES= and ZONELABELS. The ZONES option is not available in the RCHART or SCHART statements.

ZONES2
adds lines to an R or s chart that delineate zones A, B, and C for tests requested with the TESTS2= option. Related options are CZONES= and ZONE2LABELS. The ZONES2 option is available in the MRCHART, RCHART, SCHART, XRCHART, and XSCHART statements.

ZONEVALPOS=*n*
specifies the horizontal position of the ZONEVALUES= and ZONE2VALUES= labels, as described in the following table. By default, n=1.

n	Label Position
1	left-justified in subplot area
2	right-justified in subplot area
3	left-justified in right margin

ZONEVALUES
labels the primary chart zones lines with their values. If the ZONEVALUES option is specified the ZONES option is not required.

Chapter 54
Graphical Enhancements

Chapter Contents

OVERVIEW 1929

DISPLAYING STRATIFIED PROCESS DATA 1929
- Displaying Stratification in Levels of a Classification Variable 1931
- Displaying Stratification in Blocks of Observations 1932
- Displaying Stratification in Phases 1936

DISPLAYING MULTIPLE SETS OF CONTROL LIMITS 1939

DISPLAYING AUXILIARY DATA WITH STARS 1948
- Creating a Basic Star Chart 1949
- Adding Reference Circles to Stars 1950
- Specifying the Style of Stars 1952
- Specifying the Method of Standardization 1955

DISPLAYING TRENDS IN PROCESS DATA 1957
- Step 1: Preliminary Mean and Standard Deviation Charts 1959
- Step 2: Modeling the Trend 1960
- Step 3: Displaying the Trend Chart 1961

CLIPPING EXTREME POINTS 1962

LABELING AXES 1966
- Default Labels 1966
- Labeling the Horizontal Axis 1967
- Labeling the Vertical Axis 1967

SELECTING SUBGROUPS FOR COMPUTATION AND DISPLAY . . . 1970
- Using WHERE Statements 1970
- Using Switch Variables 1973

Chapter 54
Graphical Enhancements

Overview

This chapter provides details on the following topics:

- displaying process data stratified into levels using a *symbol-variable*
- displaying process data stratified into blocks using *block-variables*
- displaying process data stratified into time phases using the READPHASES= option
- displaying multiple sets of control limits using the READPHASES= and READINDEXES= options
- displaying multivariate process data using star charts
- displaying trends in process data
- clipping extreme points to create more readable charts
- labeling axes
- selecting subgroups for computation and display

The options described in this chapter can be specified in all the chart statements available in the SHEWHART procedure.

Displaying Stratified Process Data

If the data for a Shewhart chart can be classified by factors relevant to the process (for instance, machines or operators), displaying the classification on the chart can facilitate the identification of special or common causes of variation that are related to the factors. Kume (1985) refers to this type of classification as "stratification" and describes various ways to create stratified control charts.

There are important differences between stratification and subgrouping. The data must always be classified into subgroups before a control chart can be produced. Subgrouping affects how control limits are computed from the data as well as the outcome of tests for special causes (see Chapter 55, "Tests for Special Causes,"). The values of the *subgroup-variable* specified in the chart statement classify the data into subgroups. In contrast, stratification is optional and involves classification variables other than the *subgroup-variable*. Displaying stratification influences how the chart is interpreted, but it does not affect control limits or tests for special causes.

This section describes three types of variables that you can specify to create stratified control charts.

- A *symbol-variable* stratifies data into levels of a classification variable.
- The *block-variables* stratify data into blocks of consecutive observations.
- A _PHASE_ variable stratifies data into *time phases*.

You can specify any combination of these three variables. You should be careful, however, since it is possible to generate confusing charts by overusing these methods.

The data for the examples in this section consist of diameter measurements for a part produced on one of three different machines. Three subgroups, each consisting of six parts, are sampled each day, corresponding to three shifts worked each day. The data are provided in the data set PARTS, which is created by the following statements:

```
data parts;
  length machine $ 4;
  input sample machine $ day shift diamx diams;
  diamn=6;
  datalines;
 1  A386  01  1  4.32  0.39
 2  A386  01  2  4.49  0.35
 3  A386  01  3  4.44  0.44
 4  A386  02  1  4.45  0.17
 5  A386  02  2  4.21  0.53
 6  A386  02  3  4.56  0.26
 7  A386  03  1  4.63  0.39
 8  A386  03  2  4.38  0.47
 9  A386  03  3  4.47  0.40
10  A455  04  1  4.42  0.37
11  A455  04  2  4.45  0.32
12  A455  04  3  4.62  0.36
13  A455  05  1  4.33  0.31
14  A455  05  2  4.29  0.33
15  A455  05  3  4.17  0.25
16  C334  08  1  4.15  0.28
17  C334  08  2  4.21  0.33
18  C334  08  3  4.16  0.19
19  C334  09  1  4.14  0.13
20  C334  09  2  4.11  0.19
21  C334  09  3  4.10  0.27
22  C334  10  1  3.99  0.14
23  C334  10  2  4.24  0.16
24  C334  10  3  4.23  0.14
25  A386  11  1  4.27  0.28
26  A386  11  2  4.70  0.45
27  A386  11  3  4.51  0.45
28  A386  12  1  4.34  0.16
29  A386  12  2  4.38  0.29
30  A386  12  3  4.28  0.24
31  A386  15  1  4.47  0.26
32  A386  15  2  4.31  0.46
33  A386  15  3  4.52  0.33
;
run;
```

Displaying Stratification in Levels of a Classification Variable

See SHWSYM1 in the SAS/QC Sample Library

To display process data stratified into levels of a classification variable, specify the name of this variable after an equal sign (=) immediately following the *subgroup-variable* in the chart statement. The classification variable, referred to as the *symbol-variable*, must be a variable in the input data set (a DATA=, HISTORY=, or TABLE= data set). The subgroup summary statistics are classified into groups according to the levels of the *symbol-variable* and are identified on the chart with unique plotting symbols.

If you use a graphics device, you can specify the symbols with SYMBOL statements. It is recommended that you place the SYMBOL statements before the PROC SHEWHART statement. If you omit the SYMBOL statements, the procedure uses the default symbol (+) for all levels of the *symbol-variable* but plots the points for each level in a distinct color. The following example illustrates the use of a *symbol-variable* to stratify the points on an $\bar{X}$ chart according to the machine that produced the parts in each subgroup:

```
symbol1 c=black value=star       h=3.0 pct;
symbol2 c=black value=dot        h=3.0 pct;
symbol3 c=blue  value=triangle   h=3.0 pct;
title 'Control Chart for Diameter Stratified by Machine';
proc shewhart history=parts;
   xchart diam*sample=machine / stddeviations
                                symbollegend = legend1;
   label sample = 'Sample Number'
         diamx  = 'Average Diameter' ;
   legend1 frame label=('Machine');
run;
```

The symbols are specified with the SYMBOL1, SYMBOL2, and SYMBOL3 statements. The SYMBOLLEGEND= option requests a customized legend for the symbols. For more information on the LEGEND and SYMBOL statements, refer to *SAS/GRAPH Software: Reference*. The $\bar{X}$ chart, shown in Figure 54.1, reveals an effect due to MACHINE. In particular, Machine C334 is associated with a run of parts whose diameters are systematically below average, suggesting that this machine may require adjustment.

For charts produced on a line printer, you can use the SYMBOLCHARS= option to specify the characters that identify the stratification of the points. For details, see the entry for the SYMBOLCHARS= option in Chapter 53, "Dictionary of Options."

In this example, Machine A386 is associated with two different blocks of observations that are identified with a common symbol. However, a *symbol-variable* is particularly useful for situations where the stratification is not necessarily chronological or associated with blocks of consecutive groups of observations.

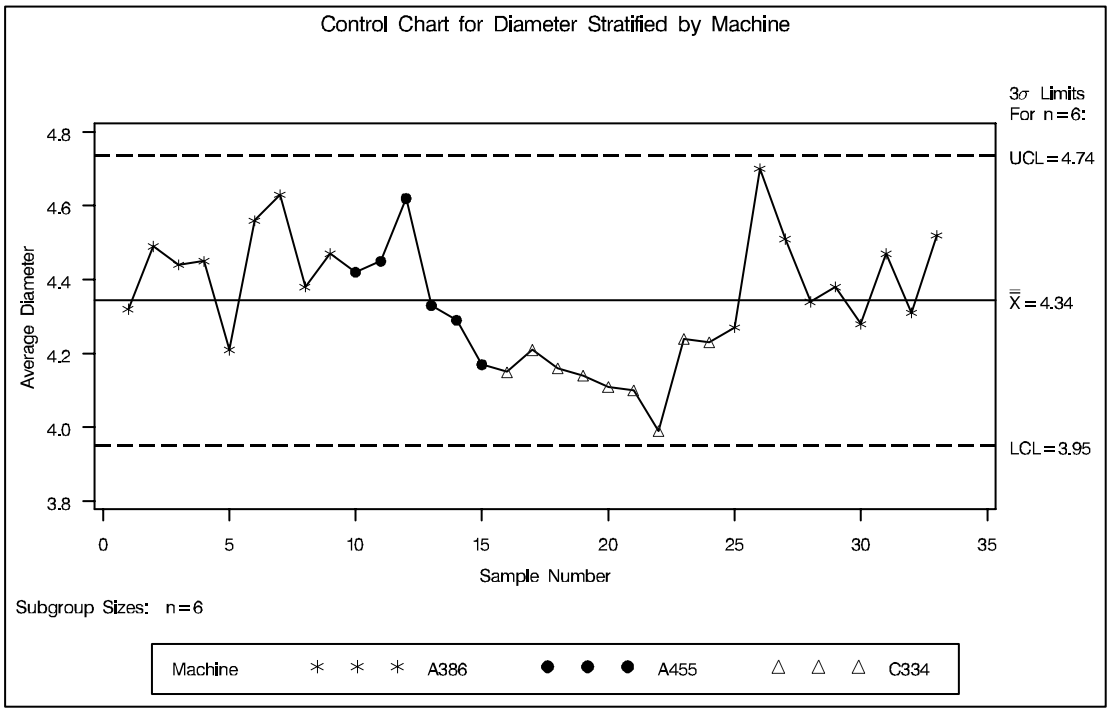


Figure 54.1. Control Chart Stratified into Levels Using Symbols

Displaying Stratification in Blocks of Observations

See SHWBLK in the SAS/QC Sample Library

To display process data stratified into blocks of consecutive observations, specify one or more *block-variables* in parentheses after the *subgroup-variable* in the chart statement. The procedure displays a legend identifying blocks of consecutive observations with identical values of the *block-variables*. The legend displays one track of values for each *block-variable*. The values are the formatted values of the *block-variable*. For example, Figure 54.2 displays a legend with a single track for MACHINE, while Figure 54.3 displays a legend with two tracks corresponding to MACHINE and DAY. You can label the tracks themselves by using the LABEL statement to associate labels with the corresponding *block-variables*; see Figure 54.4 on page 1935 for an illustration.

By default, the legend is placed above the chart as in Figure 54.2. You can control the position of the legend with the BLOCKPOS= option and the position of the legend labels with the BLOCKLABELPOS= option. See the entries in Chapter 53, "Dictionary of Options," as well as the following examples.

The *block-variables* must be variables in the input data set (a DATA=, HISTORY=, or TABLE= data set). If the input data set is a DATA= data set that contains multiple observations with the same value of the *subgroup-variable*, the values of a *block-variable* must be the same for all observations with the same value of the *subgroup-variable*. In other words, subgroups must be nested within groups determined by *block-variables*. The following statements create an $\bar{X}$ chart for the data in PARTS stratified by the *block-variable* MACHINE. The chart is shown in Figure 54.2.

```
title 'Control Chart for Diameter Stratified By Machine';
proc shewhart history=parts;
   xchart diam*sample (machine) / stddeviations
                                  nolegend ;
   label sample = 'Sample Number'
         diamx  = 'Average Diameter' ;
run;
```

The unique consecutive values of MACHINE (A386, A455, C334, and A386) are displayed in a track above the chart, and they indicate the same relationship between part diameter and machine as the previous example. Note that the track is not labeled (as in Figure 54.4), since no label is associated with MACHINE. A LABEL statement is used to provide labels for the axes.

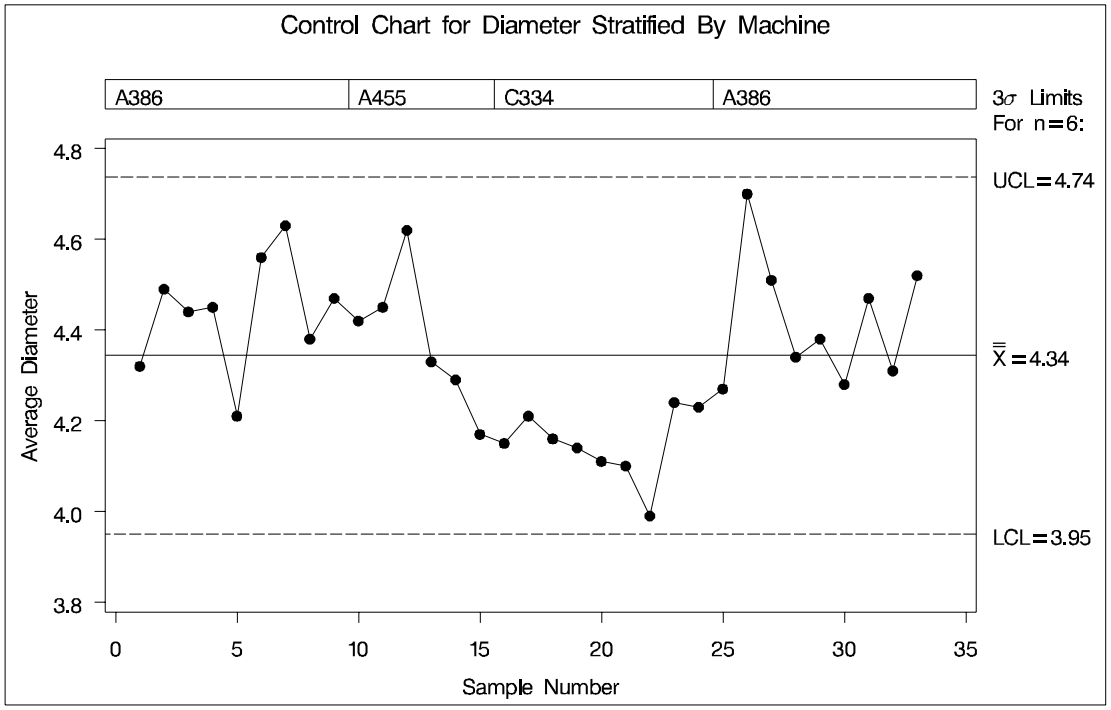


Figure 54.2. Stratified Control Chart Using a Single Block Variable

Multiple block variables. You can use multiple *block-variables* to study more than one classification factor with the same chart. The following statements create an $\bar{X}$ chart for the data in PARTS, with MACHINE and DAY as *block-variables*:

```
title 'Control Chart for Diameter Stratified By Machine and Day';
proc shewhart history=parts;
   xchart diam*sample (machine day) / stddeviations
                                      nolegend
                                      blockpos = 2;
   label sample = 'Sample Number'
         diamx  = 'Average Diameter' ;
run;
```

The chart is displayed in Figure 54.3. Specifying BLOCKPOS=2 displays the *block-variable* legend immediately above the chart, without the gap shown in Figure 54.2. The NOLEGEND option suppresses the sample size legend that appears in the lower left of Figure 54.2.

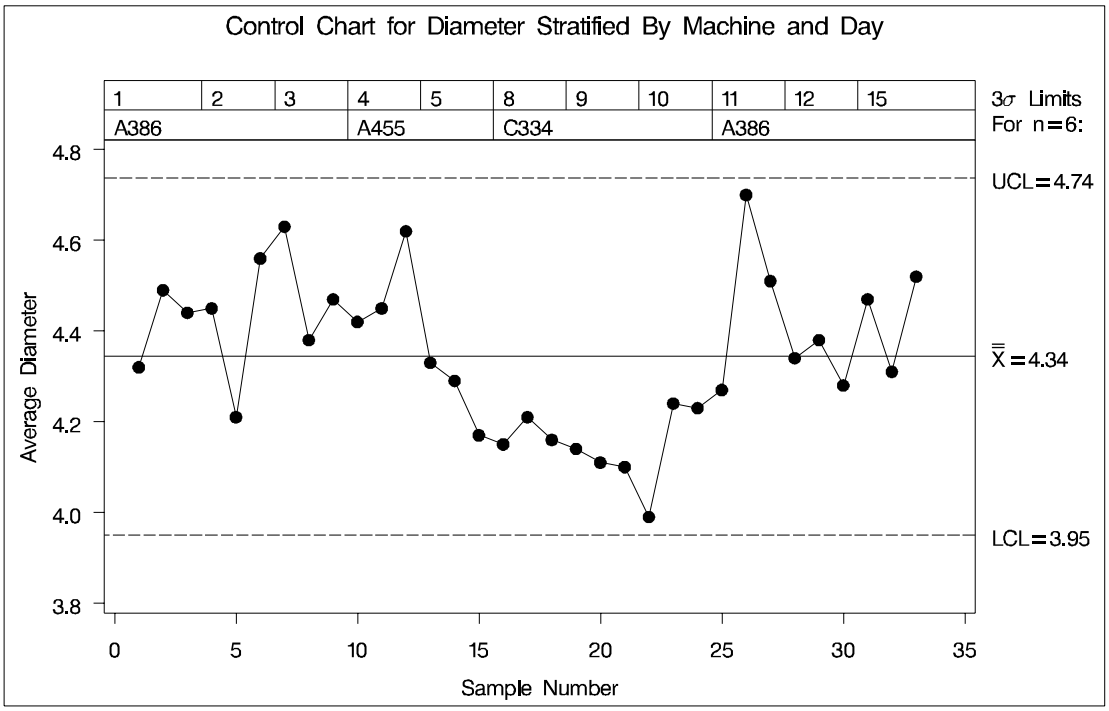


Figure 54.3. Stratified Control Chart Using Multiple Block Variables

Color fills for legend. You can use the CBLOCKVAR= option to fill the legend track sections with colors corresponding to the values of the *block-variables*. Provide the colors as values of variables specified with the CBLOCKVAR= option. These variables must be defined as character variables of length 8. The procedure matches the color variables with the *block-variables* in the order specified. Each section is filled with the color for the first observation in the block. For example, the following statements produce an $\bar{X}$ chart using a color variable named CMACHINE to fill the legend for the *block-variable* MACHINE:

```
title 'Control Chart for Diameter Stratified By Machine and Day';
proc shewhart history=parts2;
   xchart diam*sample (machine day) / stddeviations
                                      nolegend
                                      blockpos  = 3
                                      cblockvar = cmachine;
   label sample  = 'Sample Number'
         diamx   = 'Average Diameter'
         day     = 'Date of Production in June'
         machine = 'Machine in Use';
run;
```

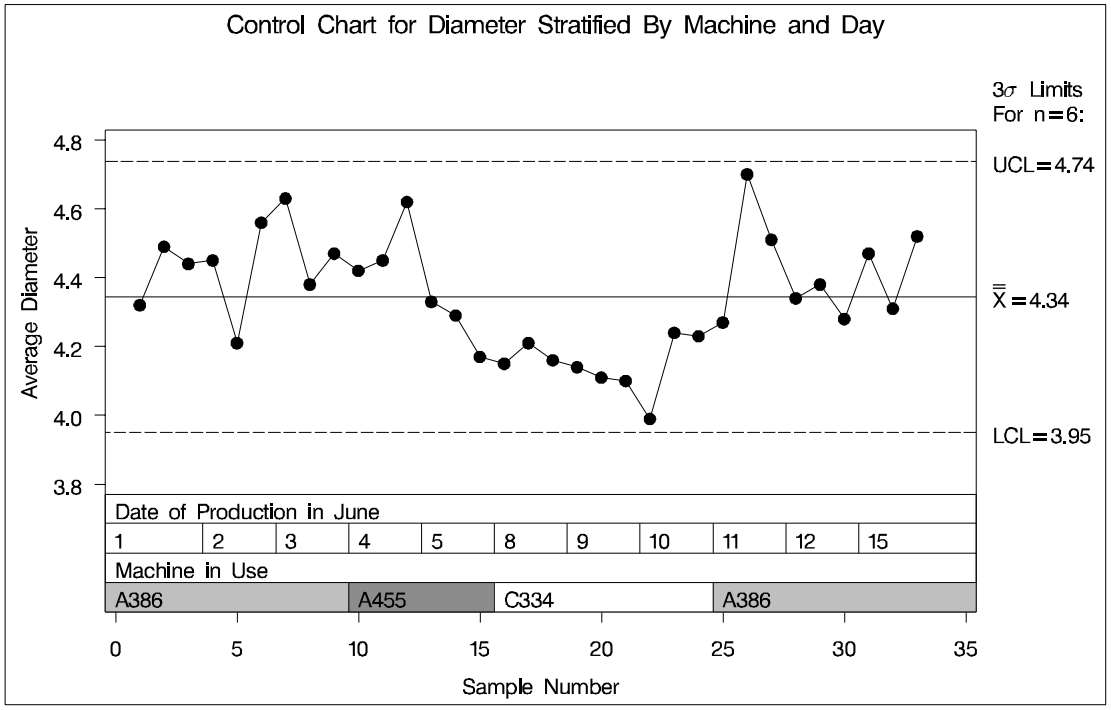


Figure 54.4. Color Fill for *Block-Variable* Legend

The sections for Machine A386 are filled with light gray, the section for Machine A455 is filled with medium gray, and the section for Machine C334 is left white. The legend track for DAY is not filled, since a second color variable was not specified with the CBLOCKVAR= option. Specifying BLOCKPOS=3 positions the legend at the bottom of the chart and facilitates comparison with the subgroup axis. The LABEL statement is used to label the tracks with the labels associated with the *block-variables*.

The following statements produce an $\bar{X}$ chart in which both legend tracks are filled:

```
title 'Control Chart for Diameter Stratified By Machine and Day';
proc shewhart history=parts3;
   xchart diam*sample (machine day) /
      stddeviations
      nolegend
      ltmargin      = 5
      blockpos      = 3
      blocklabelpos = left
      cblockvar     = (cmachine cday);
   label sample     = 'Sample Number'
         diamx      = 'Average Diameter'
         day        = 'June'
         machine    = 'Machine';
run;
```

The chart is displayed in Figure 54.5. The color values of CMACHINE are used to fill the track for MACHINE, and the color values of CDAY are used to fill the track

for DAY. Specifying BLOCKLABELPOS=LEFT displays the block variable labels to the left of the block legend. The LTMARGIN= option provides extra space in the left margin to accommodate the label *Machine*.

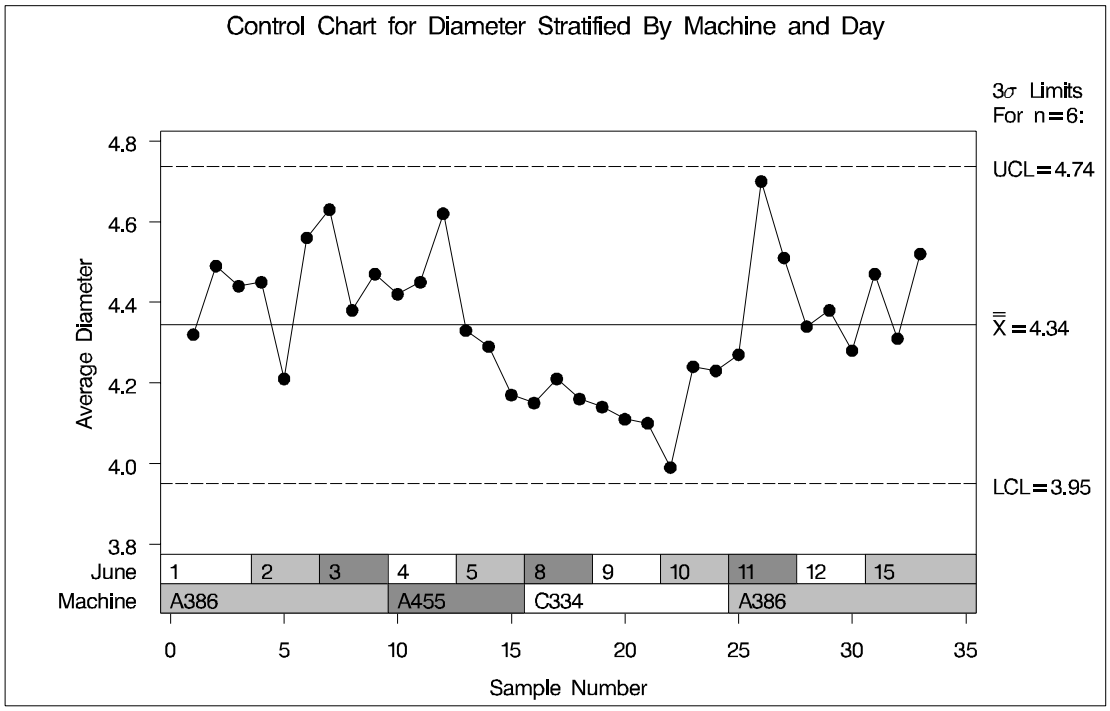


Figure 54.5. Stratified Control Chart Using Multiple Block Variables

Displaying Stratification in Phases

See SHWPHSE in the SAS/QC Sample Library

The preceding section describes the use of *block-variables* to display blocks of consecutive observations that correspond to changes in factors such as machines, shifts, and raw materials. This section describes the use of a *_PHASE_ variable* to display phases of consecutive observations (as in Figure 54.6). Although the terms *block* and *phase* have similar meanings, there are differences in the two methods:

- You can provide only one *_PHASE_ variable*, whereas you can specify multiple *block-variables*.
- You can display distinct control limits for each phase (see page 1939) but not for each block.
- Different sets of graphical options are available for identifying blocks and phases.

To display phases, your input data set must include a character variable named _PHASE_ of length 48 or less, and you must specify the READPHASES= option in the chart statement. (If your data set does not include a variable named _PHASE_, you can temporarily rename another character variable to _PHASE_, as illustrated

by the following statements.) The procedure classifies the data into phases (groups of consecutive observations with the same value of _PHASE_) and reads only those observations whose _PHASE_ value matches one of the values specified with the READPHASES= option.

You can identify and highlight the phases with various options, as illustrated by the following statements, which produce the chart shown in Figure 54.6. The PHASELEGEND option displays a legend with the _PHASE_ values, and the CPHASELEG= option specifies the color of the legend text. The PHASEREF option delineates the phases with vertical reference lines. The CFRAME= option fills the framed areas for the phases with different colors.

```
title 'Control Chart for Diameter Stratified by Machine';
proc shewhart history=parts(rename=(machine=_phase_));
   xchart diam*sample /
      stddeviations
      readphases = ('A386' 'A455' 'C334' 'A386')
      cframe     = ( ligr   megr   dagr   ligr )
      phaselegend
      cphaseleg  = black
      phaseref
      nolegend;
   label  sample = 'Sample Number'
          diamx  = 'Average Diameter';
run;
```

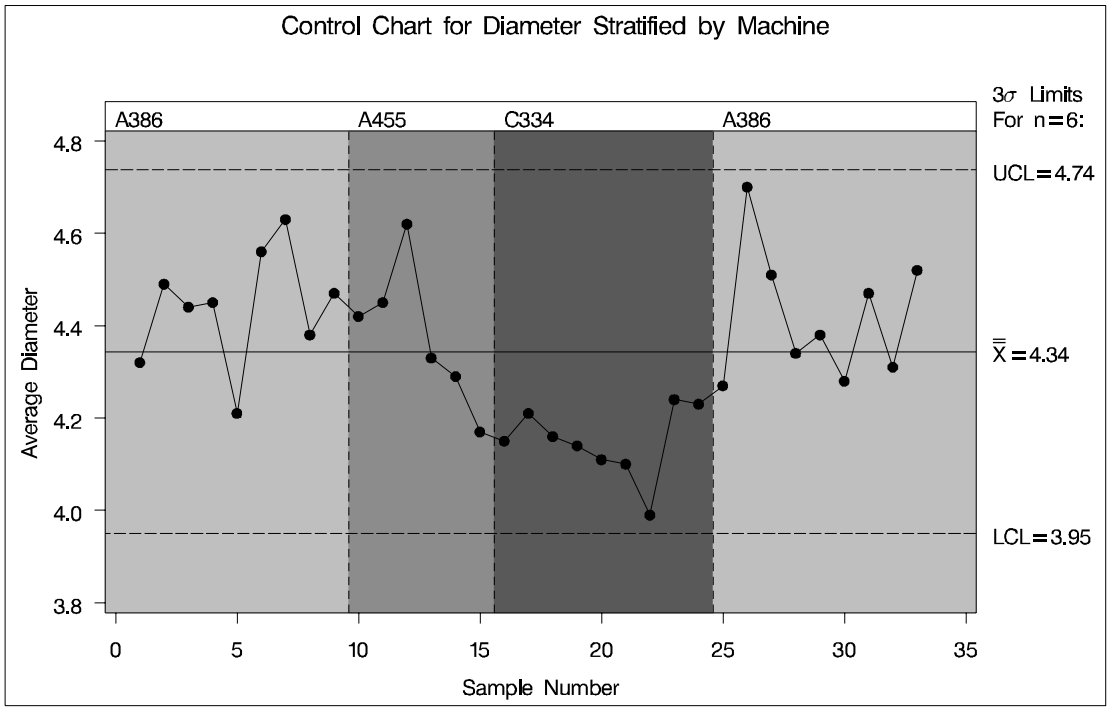


Figure 54.6. Control Chart Stratified by Phases

Note that the data set PARTS does not contain a variable named _PHASE_, so the variable MACHINE is renamed as _PHASE_ for the duration of the procedure step.

The observations read from PARTS are those whose value of MACHINE matches one of the values listed with the READPHASES= option in that order. Here, the value `A386` is listed twice; consequently, both groups of observations for which MACHINE equals `A386` are read.

In this example, the input data set contains a single observation for each subgroup. If your input data set is a DATA= data set that contains multiple observations with the same value of the *subgroup-variable*, the value of _PHASE_ must be the same for all observations with the same value of the *subgroup-variable*. Thus, in general, subgroups must be nested within phases.

Recall that the horizontal axis scale is determined by the *subgroup-variable* (see "Subgroup Variables" on page 1771). If your *subgroup-variable* is numeric, this scale is continuous; consequently, you should select phases that are reasonably contiguous in order to avoid large empty gaps in your chart. For instance, if you were to specify

```
readphases = ('A386' 'A455' 'A386')
```

in the preceding XCHART statement, there would be a gap between the 15^{th} and 25^{th} points (these points would be connected unless you specified the PHASEBREAK option). You can avoid gaps by specifying a character *subgroup-variable** for which a discrete horizontal axis scale will be displayed.

Note that the values listed in the READPHASES= option must be listed in the same order as they occur in the input data set. Thus, in order to display all the observations in the data set PARTS, `A386` must be listed as both the first and last value. An alternative method for selecting all the phases from your input data is to specify READPHASES=ALL, as described in the next section.

The control limits shown in Figure 54.6 are computed from the data and are, therefore, the same across all phases. More generally, you can display a distinct set of control limits for each phase. To do so, you must provide the control limits in a LIMITS= data set and specify the READINDEXES= option in addition to the READPHASES= option, as described in the next section.

*You can use the PUT function in a DATA step to create a character *subgroup-variable* from a numeric *subgroup-variable*.

Displaying Multiple Sets of Control Limits

See SHWCLMS in the SAS/QC Sample Library

This section describes the use of the READPHASES= and READINDEXES= options for creating Shewhart charts that display distinct sets of control limits for multiple phases of observations. The term *phase* refers to a group of consecutive observations in the input data set. For example, the phases might correspond to the time periods during which a new process was brought into production and then put through successive changes.

To display phases, your input data must include a character variable named _PHASE_, whose length cannot exceed 48. (If your data set does not include a variable named _PHASE_, you can temporarily rename another character variable to _PHASE_, as illustrated in the statements in Displaying Stratification in Phases. on page 1937.) Each phase consists of a group of consecutive observations with the same value of _PHASE_.

To display distinct sets of predetermined control limits for the phases, you must provide the limits in a LIMITS= data set. This data set must include a character variable named _INDEX_, whose length cannot exceed 48. This variable identifies the sets of control limits (observations) in the LIMITS= data set that are to be associated with the phases. This data set must also include a number of other variables with reserved names that begin and end with an underscore. The particular structure of a LIMITS= data set depends on the chart statement that you are using; for details, see the sections titled "LIMITS= Data Set" in the chapters for the various chart statements. In addition to specifying a LIMITS= data set, you must also specify the READINDEXES= and READPHASES= options in the chart statement.

Note: To display a *single* set of predetermined control limits with multiple phases, simply specify a LIMITS= data set in the procedure statement. If you are using Release 6.09 or an earlier release, you must also specify the READLIMITS option. The control limits are read from the first observation in the LIMITS= data for which the variable _VAR_ is equal to the name of the *process* and the variable _SUBGRP_ is equal to the name of the *subgroup-variable*. For an example, see "Reading Preestablished Control Limits" on page 1748.

This section describes the combinations of the READINDEXES= and READPHASES= options that you can specify. The examples that follow use the HISTORY= data set FLANGE listed in Figure 54.7 and the LIMITS= data set FLANLIM listed in Figure 54.8. The data in FLANGE consist of means and ranges of flange width measurements for subgroups of size five. The observations are grouped into three phases determined by the _PHASE_ values `Production`, `Change 1`, and `Change 2`. Three sets of control limits are provided in FLANLIM, corresponding to the _INDEX_ values `Start`, `Production`, and `Change 1`.

Obs	_phase_	day	sample	flwidthx	flwidthr	flwidthn
1	Production	08FEB90	6	0.97360	0.06247	5
2	Production	09FEB90	7	1.00486	0.11478	5
3	Production	10FEB90	8	1.00251	0.13537	5
4	Production	11FEB90	9	0.95509	0.08378	5
5	Production	12FEB90	10	1.00348	0.09993	5
6	Production	15FEB90	11	1.02566	0.06766	5
7	Production	16FEB90	12	0.97053	0.07608	5
8	Production	17FEB90	13	0.94713	0.10170	5
9	Production	18FEB90	14	1.00377	0.04875	5
10	Production	19FEB90	15	0.99604	0.08242	5
11	Change 1	22FEB90	16	0.99218	0.09787	5
12	Change 1	23FEB90	17	0.99526	0.02017	5
13	Change 1	24FEB90	18	1.02235	0.10541	5
14	Change 1	25FEB90	19	0.99950	0.11476	5
15	Change 1	26FEB90	20	0.99271	0.05395	5
16	Change 1	01MAR90	21	0.98695	0.03833	5
17	Change 1	02MAR90	22	1.00969	0.06183	5
18	Change 1	03MAR90	23	0.98791	0.05836	5
19	Change 1	04MAR90	24	1.00170	0.05243	5
20	Change 1	05MAR90	25	1.00412	0.04815	5
21	Change 2	08MAR90	26	1.00261	0.05604	5
22	Change 2	09MAR90	27	0.99553	0.02818	5
23	Change 2	10MAR90	28	1.01463	0.05558	5
24	Change 2	11MAR90	29	0.99812	0.03648	5
25	Change 2	12MAR90	30	1.00047	0.04309	5
26	Change 2	15MAR90	31	0.99714	0.03689	5
27	Change 2	16MAR90	32	0.98642	0.04809	5
28	Change 2	17MAR90	33	0.98891	0.07777	5
29	Change 2	18MAR90	34	1.00087	0.06409	5
30	Change 2	19MAR90	35	1.00863	0.02649	5

Figure 54.7. Listing of the HISTORY= Data Set FLANGE

Obs	_index_	_var_	_subgrp_	_type_	_limitn_	_alpha_	_sigmas_
1	Change 1	FLWIDTH	SAMPLE	ESTIMATE	5	.0026998	3
2	Production	FLWIDTH	SAMPLE	ESTIMATE	5	.0026998	3
3	Start	FLWIDTH	SAMPLE	ESTIMATE	5	.0026998	3

Obs	_lclx_	_mean_	_uclx_	_lclr_	_r_	_uclr_	_stddev_
1	0.96167	0.99924	1.03680	0	0.06513	0.13771	0.028000
2	0.93792	0.98827	1.03862	0	0.08729	0.18458	0.037530
3	0.87088	0.96803	1.06517	0	0.16842	0.35612	0.072409

Figure 54.8. Listing of the LIMITS= Data Set FLANLIM

For each of the READINDEXES= and READPHASES= options, you can specify a single value, a list of values, or the keyword ALL. You can also leave these options unspecified. Thus, there are 16 possible combinations of specifications for the two options, as explained by the following table and notes. The two most commonly encountered combinations are

- reading a single set of limits for one or more phases (see Note 1)
- reading a set of limits matched with a set of phases (see Note 4)

READINDEXES=	READPHASES=			
	Single *Value*	Multiple *Values*	*Keyword* ALL	Not Specified
Single *Value*	See Note 1	See Note 1	See Note 2	See Note 3
Multiple *Values*	See Note 9	See Note 4	See Note 2	See Note 2
Keyword ALL	See Note 5	See Note 5	See Note 6	See Note 6
Not Specified	See Note 7	See Note 7	See Note 8	See Note 8

Note 1. READPHASES=*value*|*value-list* **and READINDEXES=***value*

The only phases (groups of observations) read are those for which _PHASE_ equals one of the *values* specified with the READPHASES= option. The chart displays a single set of control limits given by the first observation in the LIMITS= data set for which _INDEX_ is equal to the READINDEXES= *value*.

For example, the following statements create a chart for the phases `Change 1` and `Change 2`, with control limits read from the second observation in FLANLIM. The chart is displayed in Figure 54.9.

```
symbol v = none;
proc shewhart history=flange limits=flanlim;
   xchart flwidth*sample /
      readphase = ('Change 1' 'Change 2')
      readindex = ('Production')
      phaseref
      phaselegend;
run;
```

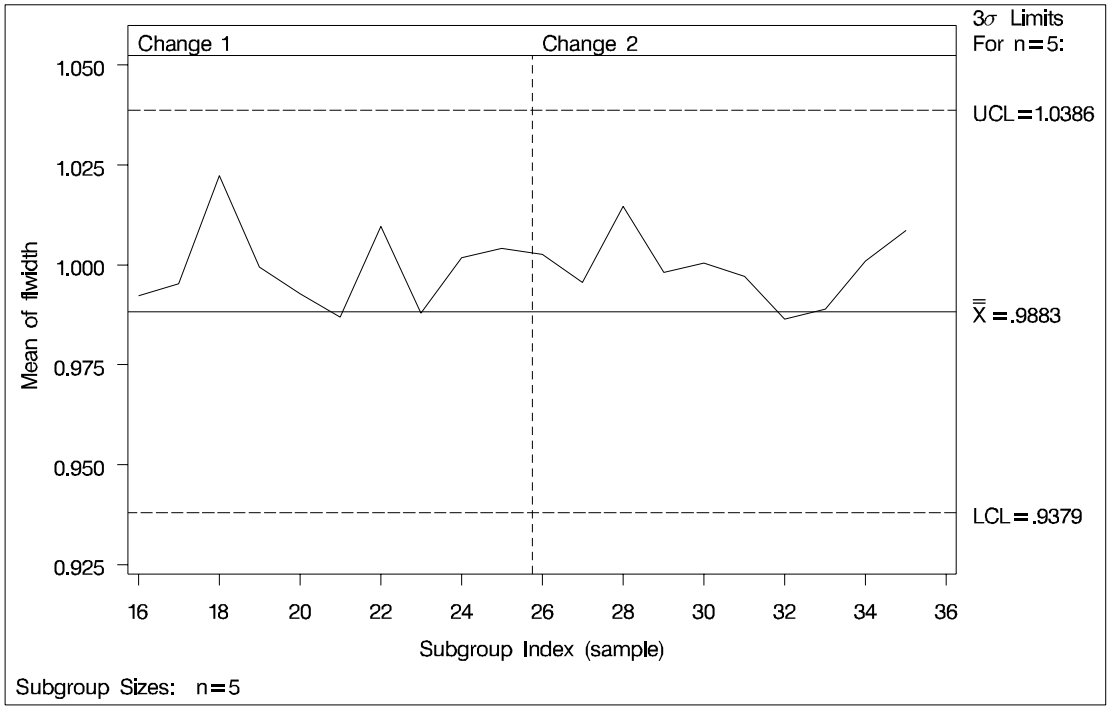


Figure 54.9. A Single Set of Control Limits for Multiple Phases

Note 2. READPHASES=ALL and READINDEXES=*value*|*value-list* or READPHASES= is omitted and READINDEXES=*value-list*

The only phases read are those for which _PHASE_ equals one of the *values* specified with the READINDEXES= option. The chart displays a different set of control limits for each phase, read from the first observation in the LIMITS= data set for which _INDEX_ is equal to the corresponding *value*.

For example, the following statements create a chart for the phases `Production` and `Change 1` with control limits read from the second and first observations in FLANLIM, respectively. The chart is displayed in Figure 54.10.

```
symbol v = none;
proc shewhart history=flange limits=flanlim;
   xchart flwidth*sample /
      readphase = all
      readindex = ('Production' 'Change 1')
      phaseref
      phaselegend;
run;
```

If you wish to specify a single set of control limits to use with all the phases, use the READINDEXES= option *without* the READPHASES= option (see Note 3).

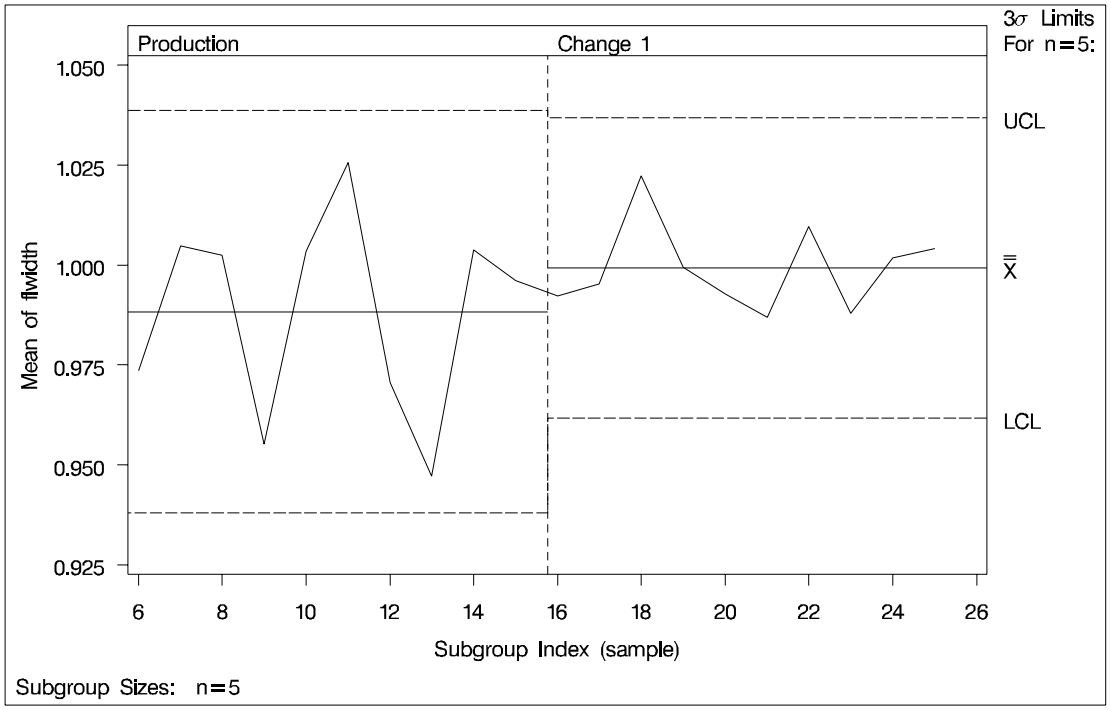


Figure 54.10. READPHASES=ALL with a List of Values for READINDEXES=

Note 3. READPHASES= is omitted and READINDEXES=*value*

All observations are read from the input data set. The chart displays a single set of control limits read from the first observation in the LIMITS= data for which _INDEX_ equals the *value*.

Note 4. READPHASES=*value-list* and READINDEXES=*value-list*

The only phases read are those for which _PHASE_ equals one of the values specified with the READPHASES= option. The chart displays a different set of control limits for each phase, given by the first observation in the LIMITS= data set for which _INDEX_ equals the READINDEXES=*value*. Control limits are matched with phases in the order listed.

For example, the following statements create a chart for the phases `Production` and `Change 1` with control limits read from the first and second observations in FLANLIM, respectively. The chart produced by these statements is identical to the chart in Figure 54.10.

```
symbol v = none;
proc shewhart history=flange limits=flanlim;
   xchart flwidth*sample /
      readphases  = ('Production' 'Change 1')
      readindexes = ('Production' 'Change 1')
      phaseref
      phaselegend;
run;
```

The order of the READINDEX=*value-list* is critical. For instance, the previous statements with READINDEXES=('Change 1' 'Production') create the chart in Figure 54.11, in which the control limits are mismatched with the phases.

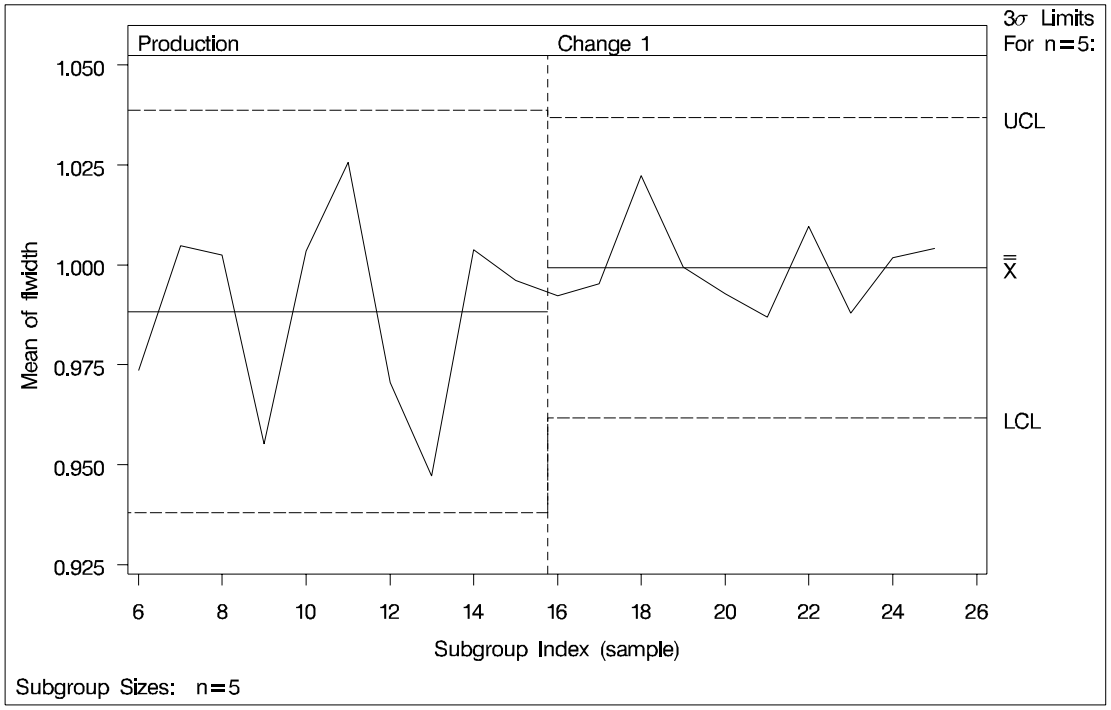


Figure 54.11. Multiple Phases with Mismatched Control Limits

Note 5. READPHASES=*value*|*value-list* and READINDEXES=ALL

The only phases read are those for which _PHASE_ equals one of the *values* specified with the READPHASES= option. The chart displays a different set of control limits for each phase, read from the first observation in the LIMITS= data set for which _INDEX_ equals the *value* corresponding to the phase.

For example, the following statements create a chart for the phases `Production` and `Change 1` with the control limits read from the second and first observations in FLANLIM, respectively:

```
proc shewhart history=flange limits=flanlim;
   xchart flwidth*sample /
      readphases  = ('Production' 'Change 1')
      readindexes = all
      phaseref
      phaselegend ;
run;
```

The chart is identical to the chart in Figure 54.10. In general, to read a set of phases with identically labeled control limits, you can specify the phases with either the READPHASES= or READINDEXES= option, and you can specify the keyword ALL with the other option.

Note 6. READPHASES=ALL and READINDEXES=ALL or READPHASES= is omitted and READINDEXES=ALL

All phases are read for which _PHASE_ is a value of _INDEX_ in the LIMITS= data set. The chart displays a different set of control limits for each phase, read from the first observation in the LIMITS= data set for which _INDEX_ equals the value of _PHASE_.

For example, the following statements create a chart for the phases `Production` and `Change 1` with control limits read from the second and first observations in FLANLIM, respectively. These two phases are read because they are the only phases in FLANGE with matching _INDEX_ values in FLANLIM. The chart is identical to that in Figure 54.10.

```
proc shewhart history=flange limits=flanlim;
   xchart flwidth*sample /
      readphase = all
      readindex = all
      phaseref
      phaselegend ;
run;
```

Note that an identical chart would be produced if you were to omit the READPHASES= option.

Note 7. READPHASES=*value*|*value-list* and READINDEXES= is omitted

The only phases read are those for which _PHASE_ equals one of the *values* specified with the READPHASES= option. The chart displays a single set of control limits read from the first observation in the LIMITS= data set for which

VAR equals the *process* and _SUBGRP_ equals the name of the *subgroup-variable* specified in the chart statement.

For example, the following statements create a chart for the phases `Production` and `Change 1` with control limits read from the first observation in FLANLIM, because this is the first observation for which _VAR_ equals `FLWIDTH` and _SUBGRP_ equals `SAMPLE`.

```
symbol v = none;
proc shewhart history=flange limits=flanlim;
   xchart flwidth*sample /
      readphases = ('Production' 'Change 1')
      phaseref
      phaselegend;
run;
```

The chart is displayed in Figure 54.12.

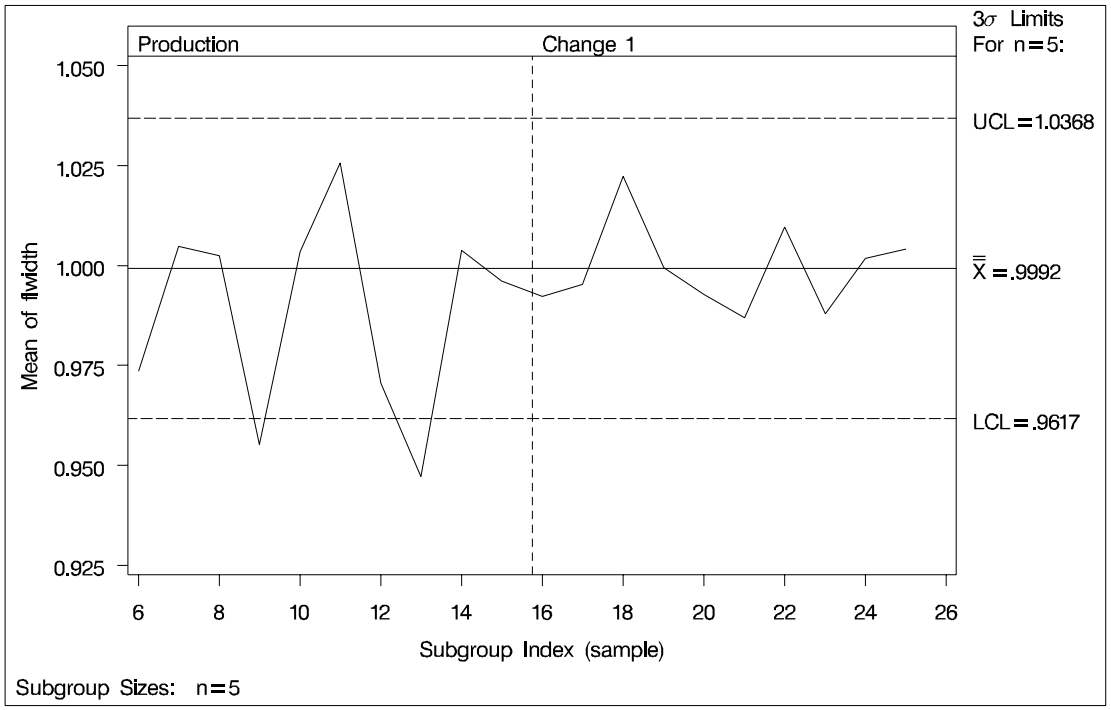


Figure 54.12. *Value-list* for READPHASES= with READINDEXES= Omitted

Note 8. READPHASES=ALL and READINDEXES= is omitted or READPHASES= is omitted and READINDEXES= is omitted

All observations are read from the input data set. The chart displays a single set of control limits read from the first observation in the LIMITS= data set for which _VAR_ equals the *process* and _SUBGRP_ equals the name of the *subgroup-variable* specified in the chart statement.

For example, the following statements create a chart for all the phases in FLANGE with control limits read from the first observation in FLANLIM,

because this is the first observation for which _VAR_ equals FLWIDTH and _SUBGRP_ equals SAMPLE:

```
symbol v = none;
proc shewhart history=flange limits=flanlim;
   xchart flwidth*sample /
      readphase = all
      phaseref
      phaselegend;
run;
```

The chart is shown in Figure 54.13. Note that an identical chart would be produced if you were to omit the READPHASES= option (except that the phase reference lines and phase legends would be omitted).

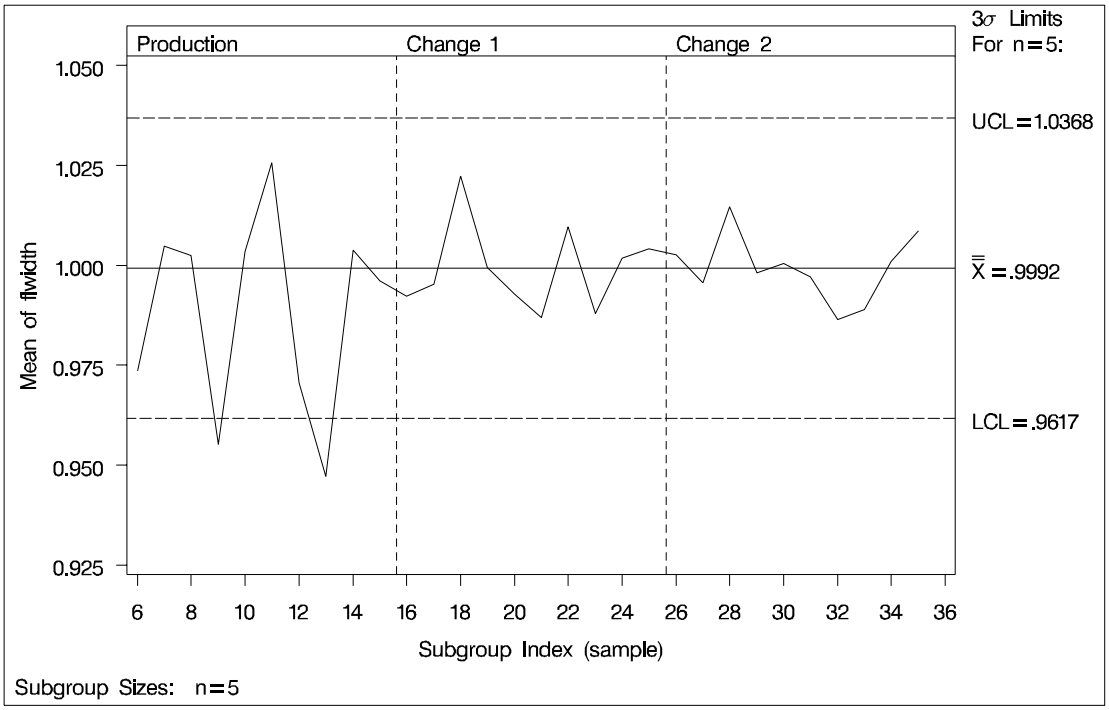


Figure 54.13. READPHASES=ALL with READINDEXES= Omitted

Note 9. READPHASES=*value* and READINDEXES=*value-list*

The procedure generates an error message.

The following tables summarize the various combinations of the READPHASES= and READINDEXES= options that you can specify.

Table 54.1. READINDEXES=*index-value*

READPHASES=	Phases Displayed	Control Limits Displayed
phase-value	_PHASE_ = *phase-value*	_INDEX_ = *index-value*
phase-value list	_PHASE_ = *phase-value list*	_INDEX_ = *index-value*
Keyword ALL	_PHASE_ = *index-value*	_INDEX_ = *index-value*
Not Specified	All phases	_INDEX_ = *index-value*

Table 54.2. READINDEXES=*index-value list*

READPHASES=	Phases Displayed	Control Limits Displayed
phase-value	No chart displayed	No chart displayed
phase-value list	_PHASE_ = *phase-value list*	_INDEX_ = *index-value list* with control limits matched to phases in the order listed
Keyword ALL	_PHASE_ = *index-value list*	_INDEX_ = *index-value list*
Not Specified	_PHASE_ = *index-value list*	_INDEX_ = *index-value list*

Table 54.3. READINDEXES=ALL

READPHASES=	Phases Displayed	Control Limits Displayed
phase-value	_PHASE_ = *phase-value*	_INDEX_ = *phase-value*
phase-value list	_PHASE_ = *phase-value list*	_INDEX_ = *phase-value list*
Keyword ALL	_PHASE_ = _INDEX_	_INDEX_ = _PHASE_
Not Specified	_PHASE_ = _INDEX_	_INDEX_ = _PHASE_

Table 54.4. READINDEXES= Not Specified

READPHASES=	Phases Displayed	Control Limits Displayed
phase-value	_PHASE_ = *phase-value*	First LIMITS= observation for which _VAR_=*process* name and _SUBGRP_=*subgroup-variable* name
phase-value list	_PHASE_ = *phase-value list*	same as previous entry
Keyword ALL	All phases	same as previous entry
Not Specified	All phases	same as previous entry

Displaying Auxiliary Data with Stars

See SHWSTR1 in the SAS/QC Sample Library

In many control chart applications, it is useful to relate the variation of the process to other variables that are being observed simultaneously with the variable that is charted. You can use the features described here to represent auxiliary multivariate data with stars (polygons) that are superimposed on the control chart. See Figure 54.16 on page 1950 for an illustration.

This display, referred to here as a *star chart*, enables you to analyze a process with a control chart while visualizing other quantities such as environmental variables, experimental control variables, or other process variables. The control chart itself can be a standard Shewhart chart, a moving average chart (such as an EWMA chart), or a cumulative sum control chart.

The examples in this section use the HISTORY= input data set PAINT (listed in Figure 54.14) and the LIMITS= data set PAINTLIM (listed in Figure 54.15). The data in PAINT consist of the subgroup means, ranges, and sample size (PINDEXX, PINDEXR, and PINDEXN) for an index of paint quality that was monitored on an hourly basis, with six auxiliary variables that were measured simultaneously: thickness, gloss, defects, dust, humidity, and temperature.

hour	pindexx	pindexr	pindexn	thick	gloss	defects	dust	humid	temp
1	5.8	3.0	5	0.2550	0.6800	0.2550	0.2125	0.1700	0.5950
2	6.2	2.0	5	0.2975	0.5950	0.0850	0.1700	0.2125	0.5525
3	3.7	2.5	5	0.3400	0.3400	0.4250	0.2975	0.2550	0.2125
4	3.2	6.5	5	0.3400	0.4675	0.3825	0.3485	0.2125	0.2125
5	4.7	0.5	5	0.5100	0.4250	0.5950	0.4080	0.5100	0.4675
6	5.2	3.0	5	0.5100	0.3400	0.6800	0.5525	0.5525	0.5525
7	2.6	2.0	5	0.4250	0.0425	0.8500	0.5355	0.5525	0.2550
8	2.1	1.0	5	0.3400	0.0170	0.8075	0.5950	0.5950	0.1700

Figure 54.14. Listing of the HISTORY= Data Set PAINT

	_	_		_	_						_
		s		l	s						s
		u	_	i	i	_	_	_	_	_	t
	_	b	t	m	g	l	m	u	l	u	d
	v	g	y	i	m	c	e	c	c	c	d
O	a	r	p	t	a	l	a	l	l	l	e
b	r	p	e	n	s	x	n	x	r	r	v
s	_	_	_	_	_	_	_	_	_	_	_
1	pindex	hour	estimate	5	3	2.395	3.875	5.355	0	2.5625	5.4184	1.10171

Figure 54.15. Listing of the LIMITS= Data Set PAINTLIM

The basic variable analyzed with the control chart (in this case, paint index) is referred to as the *process*. The auxiliary variables (in this case, thickness, gloss, defects, dust, humidity, and temperature) are referred to as *vertex variables*, because their values are represented by the vertices of the stars. A star chart can reveal relationships between

the process and the vertex variables, and it can reveal relationships among the vertex variables.

You can create star charts for any number of vertex variables. However, the resolution of your graphics device and the number of subgroups per page will limit your ability to distinguish the vertices of the stars. A practical upper limit is twelve vertex variables.

You can specify star options in all chart statements of the SHEWHART procedure except the BOXCHART statement. You can use these options to

- specify the style of the star
- add reference circles to indicate limits of variation for the stars
- add a legend identifying the relationship between vertices and vertex variables
- label the vertices
- specify colors and line types for individual stars
- specify the size of the stars
- specify different methods of standardization for the vertex variables

The star options apply only with control charts created with high-resolution graphics devices.

NOTE: A star chart is *not* the same as a multivariate control chart or a T^2 chart. A star chart is simply a univariate control chart enhanced with stars that represent auxiliary multivariate data. A multivariate control chart displays summary statistics (such as T^2) and control limits determined for a number of processes simultaneously. For an example of a multivariate control chart, see Figure 56.31 on page 2036. Figure 56.32 on page 2037 displays a multivariate control chart in which the principal components of the T^2 statistic are displayed with stars.

Creating a Basic Star Chart

The following statements create the star chart shown in Figure 54.16:

See SHWSTR1 in the SAS/QC Sample Library

```
symbol v = none;
title 'Variables Related to Paint Index';
proc shewhart history=paint limits=paintlim;
   xchart pindex*hour /
      nolegend
      starvertices = (thick gloss defects dust humid temp);
run;
```

This chart is essentially an $\bar{X}$ chart for paint index. However, the chart also provides information about thickness, gloss, defects, dust, humidity, and temperature. These six variables are represented by the vertices of the stars, as indicated by the legend at the bottom of the chart. By default, the legend uses a clock representation for the vertices; for instance, dust corresponds to the vertex at the six o'clock position.

The stars are centered at the points for average paint index, and the distance from the center to a vertex represents the standardized value of the variable corresponding to

the vertex. The star chart reveals that relatively high values of gloss (two o'clock) and temperature (ten o'clock) are associated with high out-of-control averages for paint index. Likewise, relatively high values of defects (four o'clock) and humidity (eight o'clock) are associated with low out-of-control averages for paint index. The star shapes reveal similarities in the data for runs 1 and 2, runs 3 and 4, runs 5 and 6, and runs 7 and 8.

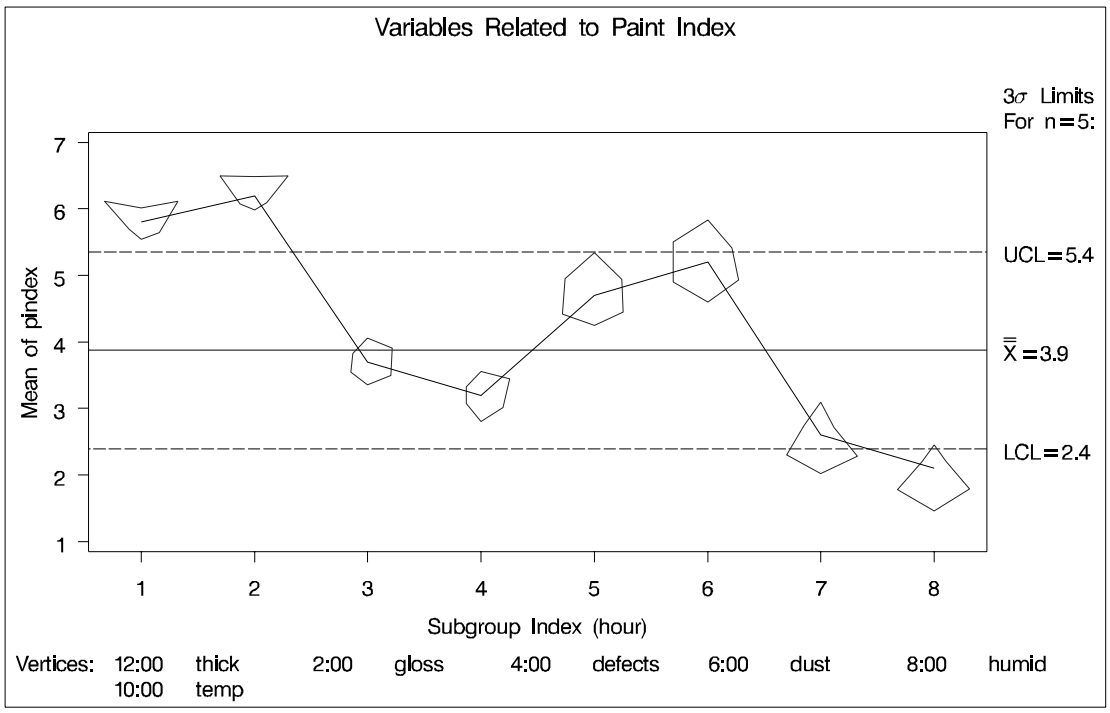


Figure 54.16. A Basic Star Chart

Adding Reference Circles to Stars

See SHWSTR1 in the SAS/QC Sample Library

You can add reference circles to a star chart to represent limits of variation for the vertex variables. The following statements add two special reference circles, called the *inner circle* and the *outer circle*, to the star chart in Figure 54.16:

```
symbol v = none;
title 'Variables Related to Paint Index';
proc shewhart history=paint limits=paintlim;
   xchart pindex*hour /
      nolegend
      starvertices = (thick gloss defects dust humid temp)
      starcircles  = 0.0 1.0
      lstarcircles = 1   2
      starstart    = '1:00'T ;
run;
```

The star chart shown in Figure 54.17 displays the two reference circles centered about each point. The STARCIRCLES= value 0.0 requests the *inner circle*, and the value

1.0 requests the *outer circle*. Whether or not they are displayed, these circles are always associated with each star.

The interpretation of the inner and outer circles depends on the method used to standardize the vertex variables. By default (as in this example), the data for each vertex variable are standardized by the range of the variable values taken across subgroups. That is, the inner circle represents the minimum value, and the outer circle represents the maximum value. You can specify other methods of standardization (see "Specifying the Method of Standardization" on page 1955).

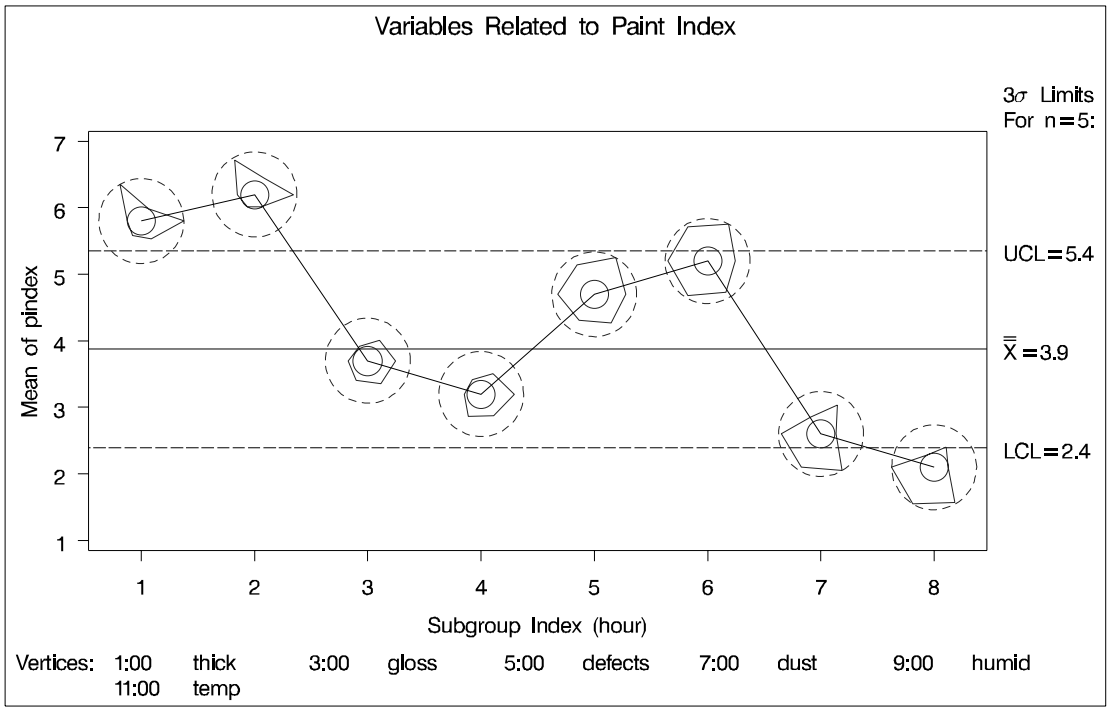


Figure 54.17. Star Chart with Inner and Outer Circles Added

Note that the STARCIRCLES= option does not specify the physical radius of a reference circle. Instead, this option specifies the radius relative to the radii of the inner and outer circles. Thus, specifying STARCIRCLES=0.0 always displays the inner circle, and specifying STARCIRCLES=1.0 always displays the outer circle. Specifying STARCIRCLES=0.5 displays a reference circle halfway between the inner and outer circles. You can specify the physical radii (in percent screen units) of the inner and outer circles using the STARINRADIUS= and STAROUTRADIUS= options. In the preceding statements, the LSTARCIRCLES= option specifies line types (1=solid and 2=dashed) for the inner and outer circles. You can also use the WSTARCIRCLES= option to control the thickness of the circles.

The STARSTART= option gives the starting position for the first vertex variable listed. In the preceding example, this option specifies that the vertex corresponding to (THICK) is to be positioned at one o'clock. The remaining vertices are uniformly spaced clockwise and correspond to the vertex variables in the order listed with the STARVERTICES= option.

For more information about the star options, see the appropriate entries in Chapter 53, "Dictionary of Options."

Specifying the Style of Stars

See SHWSTR2 in the SAS/QC Sample Library

The following statements create star charts for paint index using different styles for the stars specified with the STARTYPE= option:

```
symbol v = none;
title 'Variables Related to Paint Index';
proc shewhart history=paint limits=paintlim;
   xchart pindex * hour /
      nolegend
      starvertices = ( thick gloss defects dust humid temp )
      starstart    = '1:00'T
      startype     = wedge;

   xchart pindex * hour /
      nolegend
      starvertices = ( thick gloss defects dust humid temp )
      starstart    = '1:00'T
      startype     = radial;

   xchart pindex * hour /
      nolegend
      starvertices = ( thick gloss defects dust humid temp )
      starstart    = '1:00'T
      startype     = spoke;

   xchart pindex * hour /
      nolegend
      starvertices = ( thick gloss defects dust humid temp )
      starstart    = '1:00'T
      startype     = corona;
run;
```

The charts are shown in Figure 54.18, Figure 54.19, Figure 54.20, and Figure 54.21. The default style for the stars is STARTYPE=POLYGON, which is illustrated in Figure 54.16 and Figure 54.17. For more information, see the entry for the STARTYPE= option in Chapter 53, "Dictionary of Options."

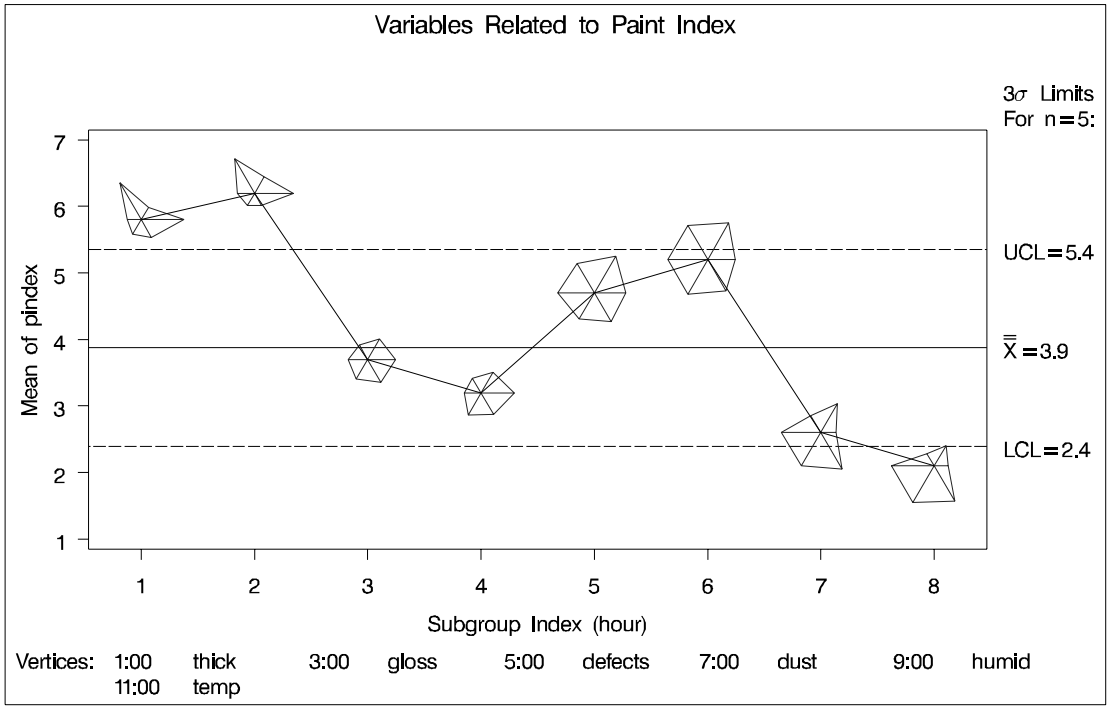


Figure 54.18. Star Chart Using STARTYPE=WEDGE

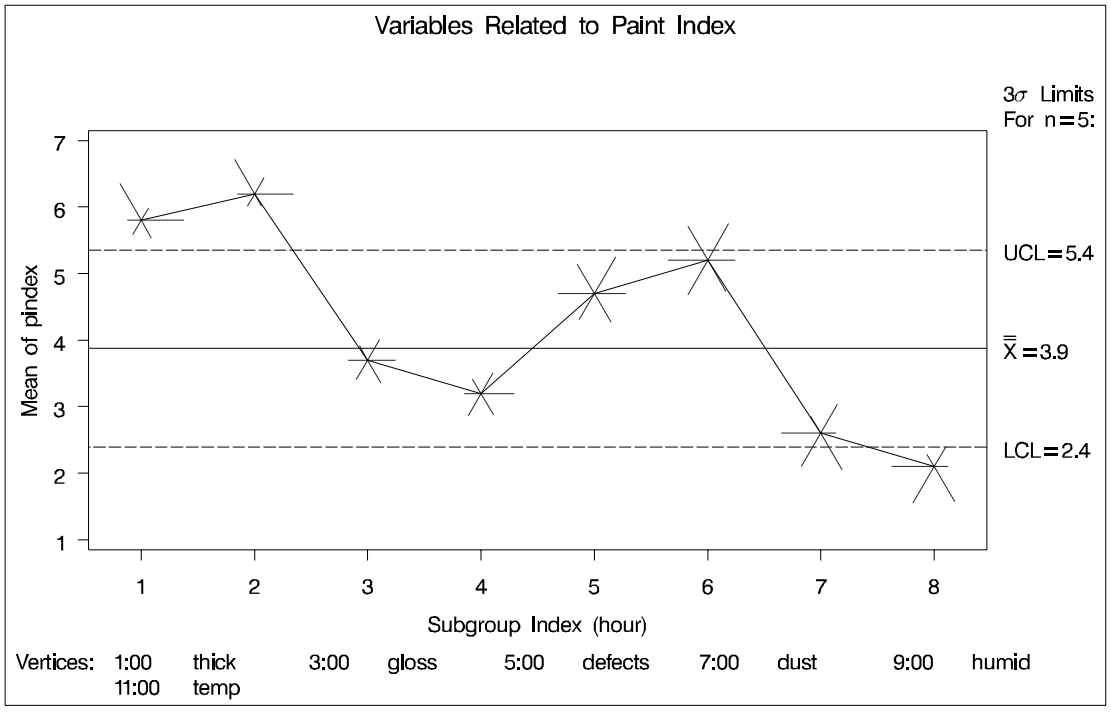


Figure 54.19. Star Chart Using STARTYPE=RADIAL

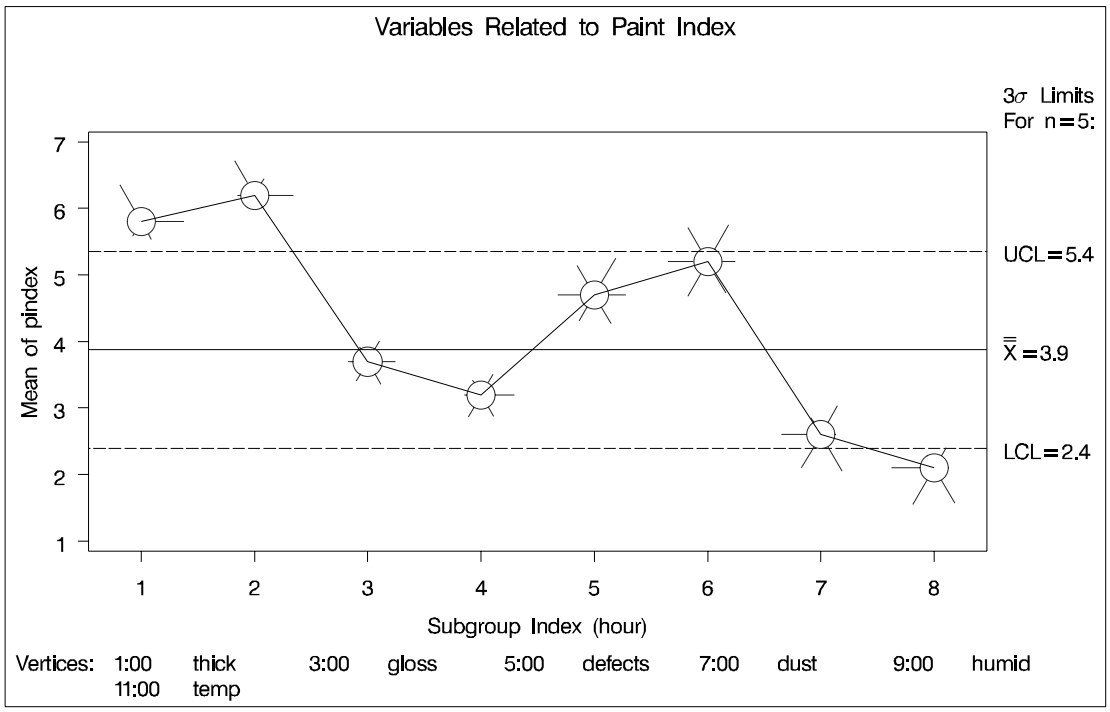


Figure 54.20. Star Chart Using STARTYPE=SPOKE

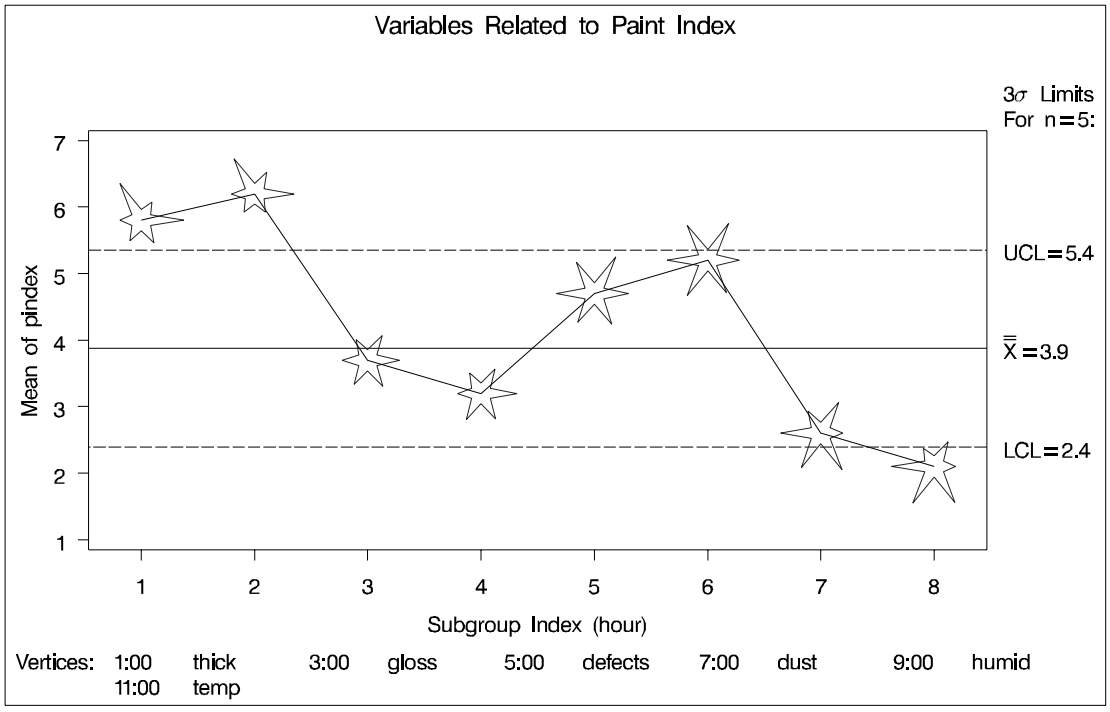


Figure 54.21. Star Chart Using STARTYPE=CORONA

Specifying the Method of Standardization

See SHWSTR3 in the SAS/QC Sample Library

In the previous examples in this section, the default method of standardization (based on ranges) is used for all six vertex variables. You can specify alternative methods with the STARSPECS= option. For example, specifying STARSPECS=3 standardizes each vertex variable so that the inner circle corresponds to three standard deviations below the mean and the outer circle corresponds to three standard deviations above the mean (that is, the circles represent 3σ limits). Specifying STARSPECS=k requests circles corresponding to $k\sigma$ limits, and specifying STARSPECS=0 requests the default method.

In some applications, it may be necessary to use distinct methods of standardization for the vertex variables. You can do this by creating an input SAS data set that provides the method for each vertex variable and specifying this data set with the STARSPECS= option.

The following statements create a data set named MYSPECS that specifies standardization methods for the vertex variables used in the previous examples:

```
data myspecs;
   length _var_     $8
          _label_   $16 ;
   input  _var_ _label_ _lspoke_ _sigmas_ _lsl_ _usl_ ;
datalines;
thick    Thickness    1        .     0.25  0.50
gloss    Gloss        1        .     0.10  0.60
defects  Defects      1        .     0.10  0.60
dust     Dust         2      3.0      .     .
humid    Humidity     2      0.0      .     .
temp     Temperature  2      0.0      .     .
;
run;
```

This data set contains a number of special variables whose names begin and end with an underscore.

Variable Name	Description
LABEL	label for identifying the vertex (used in conjunction with the STARLABEL= option). This must be a character variable of length 16 or less.
LSL	lower specification limit
LSPOKE	line style for spokes used with STARTYPE=RADIAL, STARTYPE=SPOKE, and STARTYPE=WEDGE
SIGMAS	multiple of standard deviations above and below the average. A value of zero specifies standardization based on the range.
USL	upper specification limit
VAR	name of vertex variable. This must be a character variable whose length is no greater than 32.

Standardization is specified with the variables _SIGMAS_, _LSL_, and _USL_, as follows:

- Since nonmissing specification limits (_LSL_ and _USL_) are provided for the variables THICK, GLOSS, and DEFECTS, the values of these variables are scaled so that the inner circle represents the lower specification limit and the outer circle represents the upper specification limit.
- Since _SIGMAS_ is equal to 3 for DUST (and since both _LSL_ and _USL_ are missing), values of DUST are scaled so that the inner circle represents three standard deviations below the mean, and the outer circle represents three standard deviations above the mean. The mean and standard deviation are calculated across all subgroups.
- Since _SIGMAS_ is equal to 0 for HUMID and TEMP (and since both _LSL_ and _USL_ are missing), values of HUMID and TEMP are scaled so that the inner circle represents the minimum and the outer circle represents the maximum. The minimum and maximum are calculated across all subgroups.

The following statements use the data set MYSPECS to create a star chart for paint index:

```
symbol v = none;
title 'Variables Related to Paint Index';
proc shewhart history=paint limits=paintlim;
   xchart pindex * hour /
      nolegend
      starvertices = ( thick gloss defects dust humid temp )
      startype     = wedge
      starcircles  = 0.0 1.0
      lstarcircles = 2   2
      starstart    = -30
      labelfont    = simplex
      starlegend   = degrees
      starspecs    = myspecs
      starlabel    = high ;
run;
```

The chart is shown in Figure 54.22. Specifying STARLEGEND=DEGREES requests a legend that identifies the vertex variables by their angles (in degrees) rather than their clock positions. Here, zero degrees corresponds to twelve o'clock, and the degrees are measured clockwise. The first vertex variable is positioned at 30 degrees, as specified with the STARSTART= option. Note that you specify the STARSTART= value as a negative number to indicate that it is in degrees.

In Figure 54.19 the vertices that exceed the outer circle are labeled with the value of the variable _LABEL_ in the STARSPECS= data set. This type of labeling is requested by specifying STARLABEL=HIGH. A font (SIMPLEX) for the labels is specified with the LABELFONT= option.

The vertices for THICK at HOUR=5, 6, and 7 are truncated, as indicated in the SAS log. The truncation value is the physical radius of an imaginary circle referred to

as the *bounding circle* that lies outside the outer circle. In general, any vertex that exceeds the bounding circle is truncated to the *bounding radius*. This is done so that unusually large vertex variable values will not result in grossly distorted stars. You can specify a different bounding radius with the STARBDRADIUS= option.

The spokes corresponding to the environmental variables DUST, HUMID, and TEMP are drawn with a dashed line style to distinguish them from the quality variables THICK, GLOSS, and DEFECTS, whose spokes are drawn with a solid line. The styles are specified by the variable _LSPOKE_. Refer to *SAS/GRAPH Software: Reference* for a complete list of line styles. If you are producing charts in color, you can also use the variable _CSPOKE_ in the STARSPECS= data set to assign colors to the spokes.

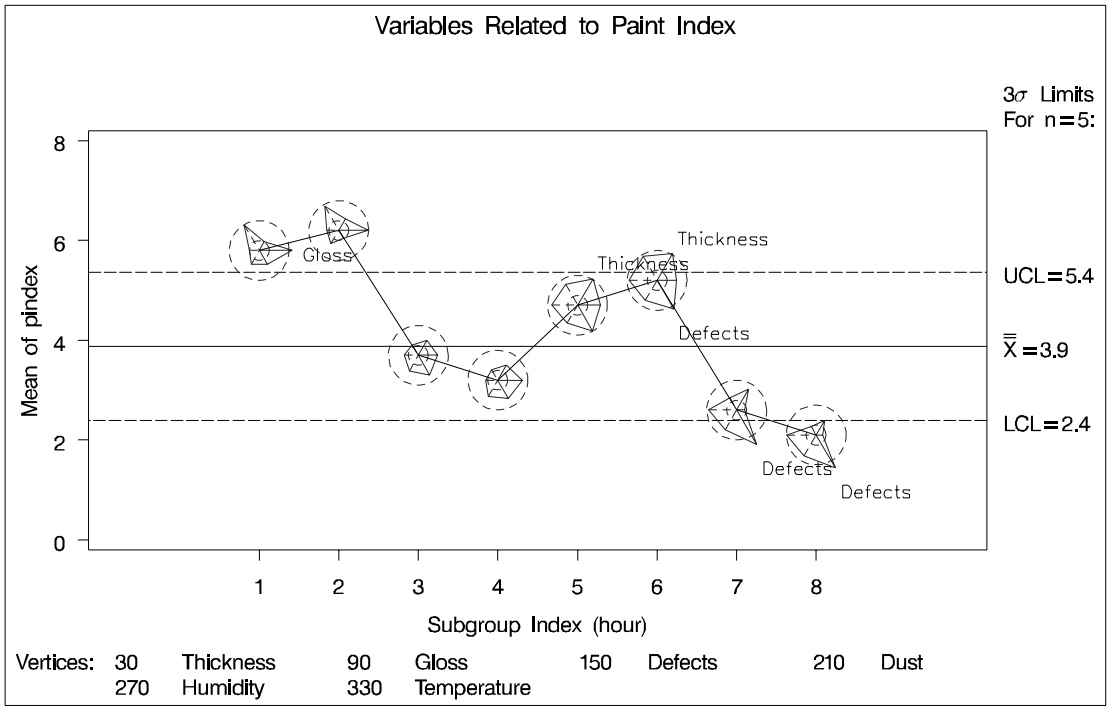


Figure 54.22. Star Chart Using STARSPECS= Specifications

For more information about the options used in this example, see the appropriate entries in Chapter 53, "Dictionary of Options."

Displaying Trends in Process Data

See SHWTREN in the SAS/QC Sample Library

Time trends due to tool wear, environmental changes, and other gradual process changes are sometimes observed in $\bar{X}$ charts. The presence of a systematic trend makes it difficult to interpret the chart because the control limits are designed to indicate expected variation strictly due to common causes.

You can use the REG procedure (or other modeling procedure) in conjunction with the SHEWHART procedure to determine whether a process with a time trend is in

control. With the REG procedure, you can model the trend and save the fitted subgroup means ($\widehat{\bar{X}}_t$) and the residual subgroup means ($\bar{X}_t - \widehat{\bar{X}}_t$) in an output data set. Then, using this data as input to the SHEWHART procedure, you can create a *trend chart*, which displays a trend plot of the fitted subgroup means together with an $\bar{X}$ chart for the residual subgroup means, thus removing the time-dependent component of the data from its random component. Having accounted for the time trend, you can decide whether the process is in control by examining the $\bar{X}$ chart.

The following example illustrates the steps used to create a trend chart for a SAS data set named TOOLWEAR that contains diameter measurements for 20 subgroup samples each consisting of eight parts:

```
data toolwear;
   input hour @;
   do i=1 to 8;
      input diameter @;
      output;
      end;
   drop i;
   datalines;
   1    10.0434   9.9427   9.9548   9.8056
        10.0780  10.0302  10.1173  10.0215
   2    10.1976   9.9654  10.0425  10.1183
        10.0963  10.1635  10.1382  10.1265
   3    10.0552  10.0695  10.2495  10.1753
        10.1268  10.1229  10.1351  10.2084
   4    10.1600  10.1378  10.2433  10.2634
        10.1808  10.1601  10.1035  10.0027
   5     9.9611  10.4322  10.1066  10.2653
        10.0310  10.1409  10.2709  10.0585
   6    10.2208  10.2298  10.2427  10.2315
        10.2048  10.2824  10.3347  10.1650
   7    10.2670  10.3793  10.2539  10.4037
        10.3281  10.1327  10.1986  10.1841
   8    10.2537  10.1981  10.2935  10.4308
        10.3195  10.3122  10.2033  10.3220
   9    10.2488  10.1866  10.3678  10.1755
        10.3225  10.2375  10.2466  10.3387
  10    10.3744  10.5221  10.2890  10.3123
        10.5134  10.3212  10.3139  10.1565
  11    10.3525  10.3237  10.4605  10.5139
        10.3650  10.1171  10.3863  10.2061
  12    10.3279  10.3338  10.1885  10.2810
        10.2400  10.3617  10.2938  10.2656
  13    10.1651  10.2404  10.1814  10.2330
        10.3094  10.3373  10.3266  10.3830
  14    10.3554  10.4577  10.5435  10.4805
        10.5358  10.4631  10.3689  10.1750
  15    10.2962  10.4221  10.3578  10.4694
        10.3465  10.4499  10.4645  10.3986
  16    10.6002  10.1924  10.3437  10.3228
        10.3438  10.3503  10.3761  10.3137
  17    10.4015  10.3592  10.3187  10.4108
        10.4834  10.4807  10.2178  10.3897
```

```
  18     10.4514   10.4492   10.3373   10.4497
         10.4197   10.3496   10.3949   10.1585
  19     10.3445   10.3310   10.4472   10.4684
         10.3975   10.2714   10.2952   10.6255
  20     10.2612   10.3824   10.4240   10.3120
         10.5744   10.4204   10.4073   10.3783
  ;
run;
```

Step 1: Preliminary Mean and Standard Deviation Charts

The following statements create $\bar{X}$ and s charts for the diameter data:

```
symbol h=3.0 pct;
title f=qcfont1 'X ' f=none 'and s Chart for Diameter';
proc shewhart data=toolwear;
   xschart diameter*hour /
      outhistory = submeans
      nolegend ;
   label diameter = 'Mean in mm';
   label hour     = 'Hour';
run;
```

The charts are shown in Figure 54.23. The subgroup standard deviations are all within their control limits, indicating the process variability is stable. However, the $\bar{X}$ chart displays a nonlinear trend that makes it difficult to decide if the process is in control. Subsequent investigation reveals that the trend is due to tool wear.

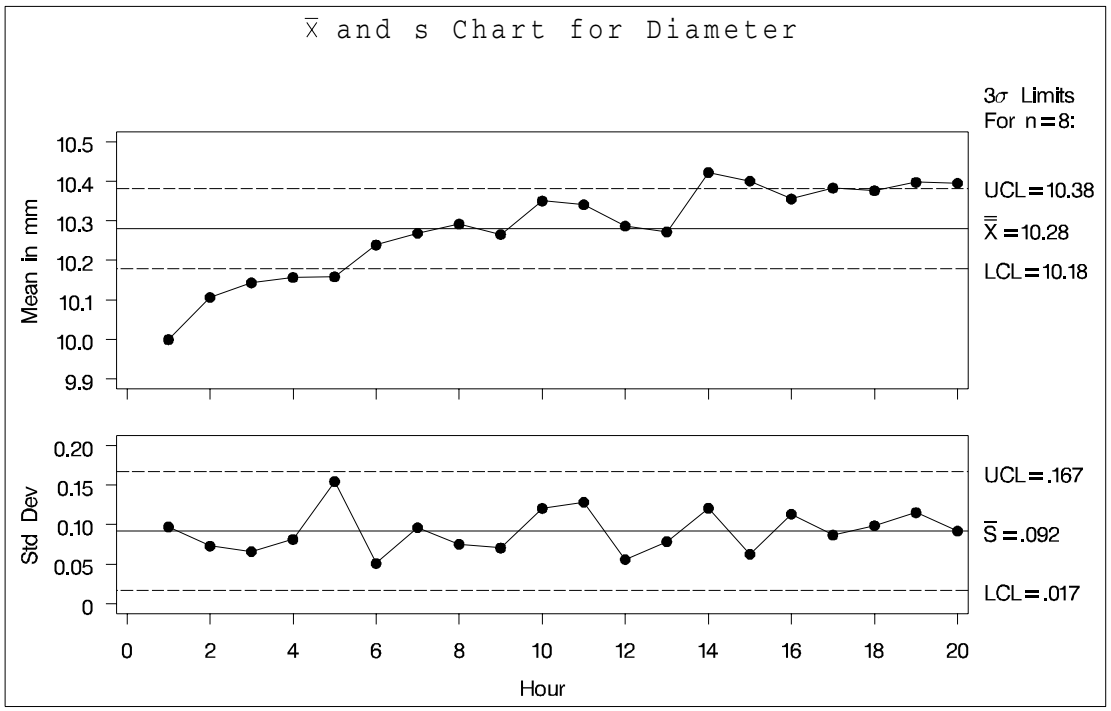


Figure 54.23. $\bar{X}$ and s Charts for TOOLWEAR Data

Note that the symbol $\bar{X}$ is displayed in the title with the special font QCFONT4, which matches the SWISS font used for the remainder of the title. See Appendix D, "Special Fonts in SAS/QC Software," for a description of the fonts available for displaying $\bar{X}$ and related symbols.

Step 2: Modeling the Trend

The next step is to model the trend as a function of hour. The $\bar{X}$ chart in Figure 54.23 suggests that the mean level of the process (saved as DIAMTERX in the OUTLIMITS= data set SUBMEANS) grows as the log of HOUR. The following statements fit a simple linear regression model in which DIAMTERX is the response variable and LOGHOUR (the log transformation of HOUR) is the predictor variable. Part of the printed output produced by PROC REG is shown in Figure 54.24.

```
data submeans;
   set submeans;
   loghour=log(hour);
run;


proc reg data=submeans ;
   model diameterx=loghour;
   output out=regdata predicted=fitted ;
run;


proc print data=regdata noobs;
run;
```

The REG Procedure
Model: MODEL1
Dependent Variable: diameterX Mean of diameter

Parameter Estimates

Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	Intercept	1	9.99056	0.02185	457.29	<.0001
loghour		1	0.13690	0.00967	14.16	<.0001

Figure 54.24. Trend Analysis for DIAMETER from PROC REG

Figure 54.24 shows that the fitted equation can be expressed as

$$\hat{\bar{X}}_t = 9.99 + 0.14 \times \log(t)$$

where $\hat{\bar{X}}_t$ is the fitted subgroup average.* A partial listing of the OUT= data set REGDATA created by the REG procedure is shown in Figure 54.25.

*Although this example does not check for the existence of a trend, you should do so by using the hypothesis tests provided by the REG procedure.

hour	diameter X	diameter S	diameter N	loghour	fitted
1	9.9992	0.09726	8	0.00000	9.9906
2	10.1060	0.07290	8	0.69315	10.0855
3	10.1428	0.06601	8	1.09861	10.1410
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
20	10.3950	0.09185	8	2.99573	10.4007

Figure 54.25. Listing of the Output Data Set REGDATA from the REG Procedure

Step 3: Displaying the Trend Chart

The third step is to create a trend chart with the SHEWHART procedure, as follows:

```
symbol v = none w=5;
title 'Trend Chart for Diameter';
proc shewhart history=regdata;
   xchart diameter*hour /
      cneedles=black
      wtrend   = 1
      trendvar = fitted
      split    = '/'
      stddevs
      nolegend;
   label diameterx = 'Residual Mean/Fitted Mean';
   label hour      = 'Hour';
run;
```

The chart is shown in Figure 54.26. The values of FITTED are plotted in the lower half of the trend chart. The upper half of the trend chart is an $\bar{X}$ chart for the residual means (DIAMTERX − FITTED). The CNEEDLES= option specifies that the residuals are to be represented by vertical bars as deviations from the central line. The $\bar{X}$ chart in Figure 54.26 shows that, after accounting for the trend, the mean level of the process is in control.

If the data are correlated in time, you can use the ARIMA or AUTOREG procedures in place of the REG procedure to remove autocorrelation structure and display a control chart for the residuals; for an example, see "Autocorrelation in Process Data" on page 2001. Another application of the TRENDVAR= option is the display of nominal values in control charts for short runs; see "Short Run Process Control" on page 2016.

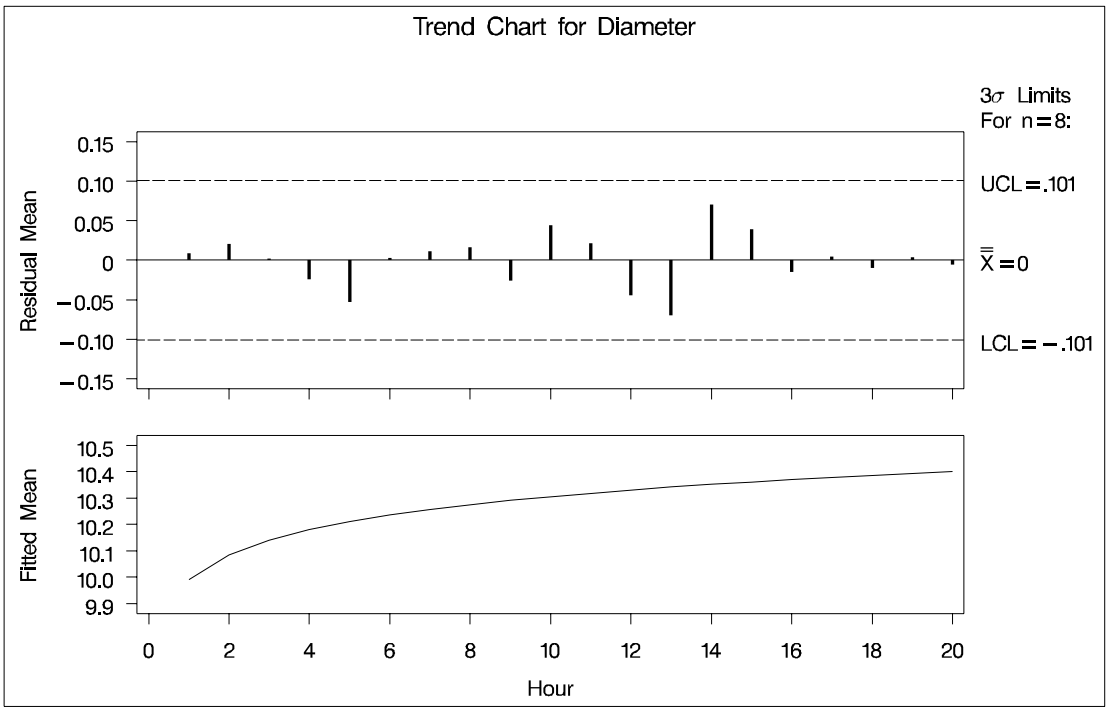


Figure 54.26. Trend Chart for Diameter Data

Clipping Extreme Points

See SHWCLIP in the SAS/QC Sample Library

In some control chart applications, the out-of-control points can be so extreme that the remaining points are compressed to a scale that is difficult to read. In such cases, you can clip the extreme points so that a more readable chart is displayed, as illustrated in the following example.

A company producing copper tubing uses $\bar{X}$ and R charts to monitor the diameter of the tubes. Based on previous production, known values of 70mm and 0.75mm are available for the mean and standard deviation of the diameter. The diameter measurements (in millimeters) for 15 batches of five tubes each are provided in the data set NEWTUBES.

```
data newtubes;
   label diameter='Diameter in mm';
   do batch = 1 to 15;
      do  i = 1 to 5;
         input diameter @@;
         output;
      end;
   end;
datalines;
69.13  69.83  70.76  69.13  70.81
85.06  82.82  84.79  84.89  86.53
67.67  70.37  68.80  70.65  68.20
71.71  70.46  71.43  69.53  69.28
```

```
71.04  71.04  70.29  70.51  71.29
69.01  68.87  69.87  70.05  69.85
50.72  50.49  49.78  50.49  49.69
69.28  71.80  69.80  70.99  70.50
70.76  69.19  70.51  70.59  70.40
70.16  70.07  71.52  70.72  70.31
68.67  70.54  69.50  69.79  70.76
68.78  68.55  69.72  69.62  71.53
70.61  70.75  70.90  71.01  71.53
74.62  56.95  72.29  82.41  57.64
70.54  69.82  70.71  71.05  69.24
;
run;
```

The following statements create the $\bar{X}$ and R charts shown in Figure 54.27 for the tube diameter:

```
symbol value=plus;
title  'Control Chart for New Copper Tubes' ;
proc shewhart data=newtubes;
   xrchart diameter*batch /
      mu0    = 70
      sigma0 = 0.75;
run;
```

Batches 2 and 7 result in extreme out-of-control points on the mean chart, and batch 14 results in an extreme out-of-control point on the range chart. The vertical axes are scaled to accommodate these extreme out-of-control points, and this in turn forces the control limits to be compressed.

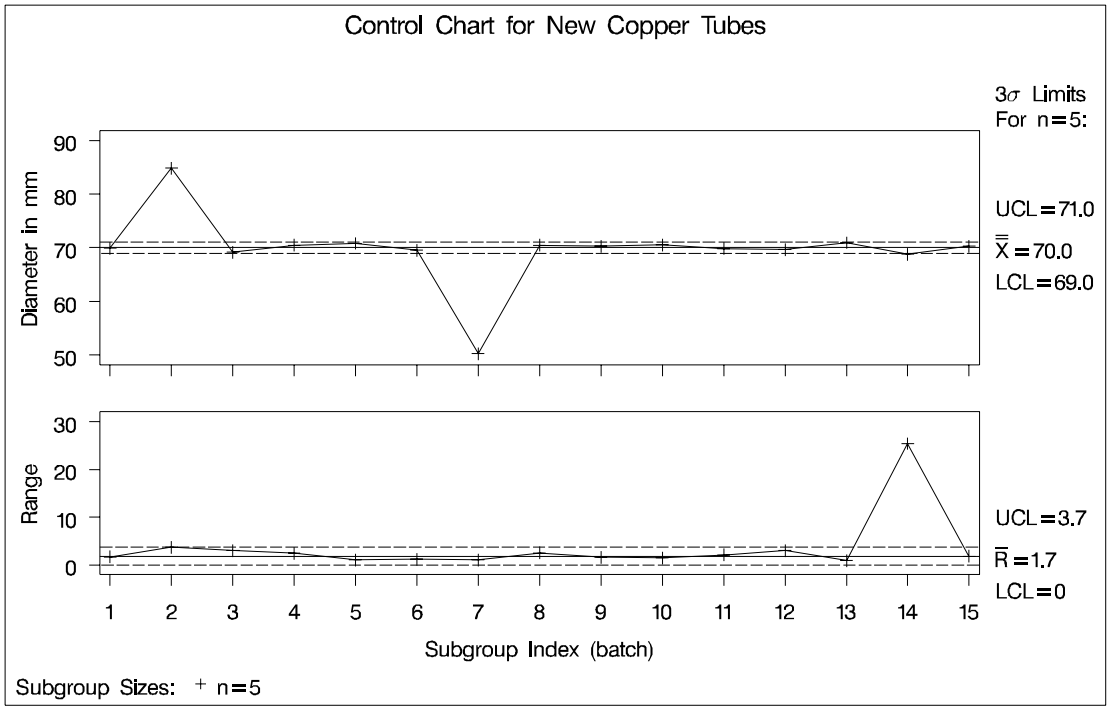


Figure 54.27. $\bar{X}$ and R Charts Without Clipping

You can request clipping by specifying the option CLIPFACTOR=*factor*, where *factor* is a value greater than one (useful values are typically in the range 1.5 to 2). Clipping is applied in two steps, as follows:

1. If a plotted statistic is greater than $y_{\max}$, it is temporarily set to $y_{\max}$, where

   $$y_{\max} = \text{LCL} + (\text{UCL} - \text{LCL}) \times \textit{factor}$$

   If a plotted statistic is less than $y_{\min}$, it is temporarily set to $y_{\min}$, where

   $$y_{\min} = \text{UCL} - (\text{UCL} - \text{LCL}) \times \textit{factor}$$

2. Axis scaling is applied to the clipped statistics. Then the $y_{\max}$ values are reset to the maximum value on the axis and the $y_{\min}$ values are reset to the minimum value on the axis.

Notes:

- Clipping is applied only to the plotted statistics and not to the statistics tabulated or saved in an output data set.
- Because the *factor* must be greater than one, clipping does not affect whether a plotted statistic is inside or outside the control limits.
- Tests for special causes are applied to the plotted statistics before they are clipped, and clipping does not affect how the tests are flagged on the chart. In some situations, however, clipping can make the patterns associated with the tests less evident on the chart.
- When primary and secondary charts are displayed, the same clipping *factor* is applied to both charts.
- A special symbol is used for clipped points (the default symbol is a square), and a legend is added to the chart indicating the number of points that were clipped.

The following statements create $\bar{X}$ and R charts, shown in Figure 54.28, that use a clipping factor of 1.5:

```
symbol value=plus;
title  'Control Chart for New Copper Tubes' ;
proc shewhart data=newtubes;
   xrchart diameter*batch /
      mu0        = 70
      sigma0     = 0.75
      clipfactor = 1.5;
run;
```

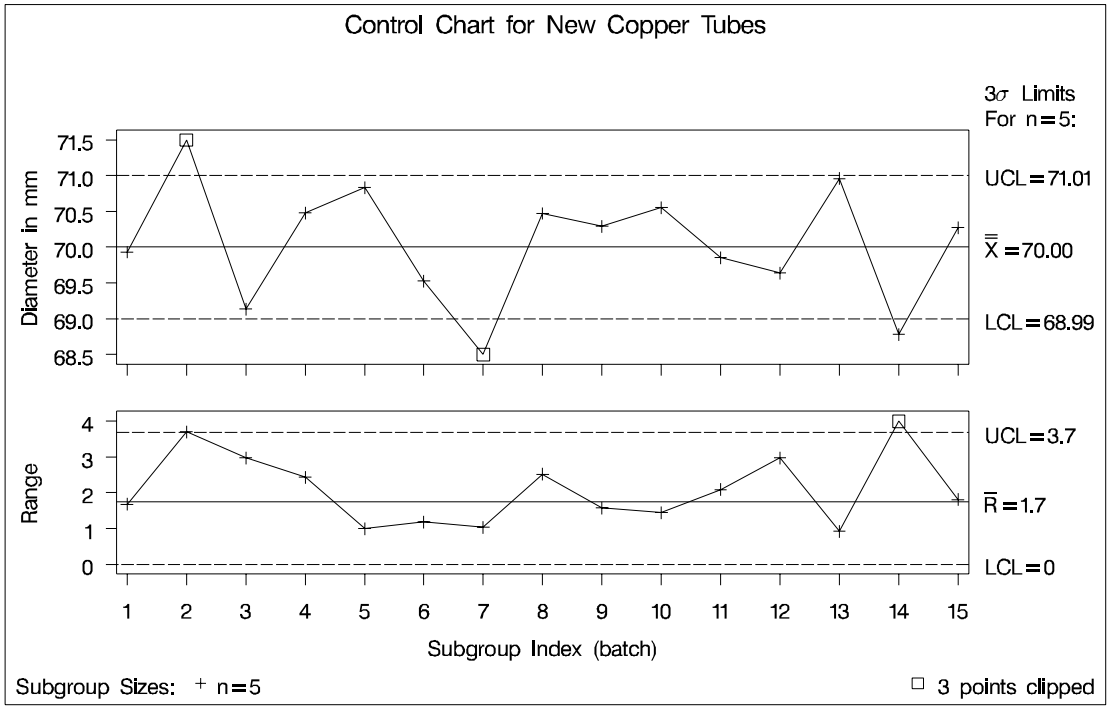


Figure 54.28. $\bar{X}$ and R Charts with Clip Factor of 1.5

In Figure 54.28, the extreme out-of-control points are clipped making the points plotted within the control limits more readable. The clipped points are marked with a square, and a clipping legend is added at the lower right of the display.

Other clipping options are available, as illustrated by the following statements:

```
symbol value=plus;
title  'Control Chart for New Copper Tubes' ;
proc shewhart data=newtubes;
   xrchart diameter*batch /
      mu0          = 70
      sigma0       = 0.75
      clipfactor   = 1.5
      clipsymbol   = dot
      cliplegpos   = top
      cliplegend   = '# Clipped Points'
      clipsubchar = '#';
run;
```

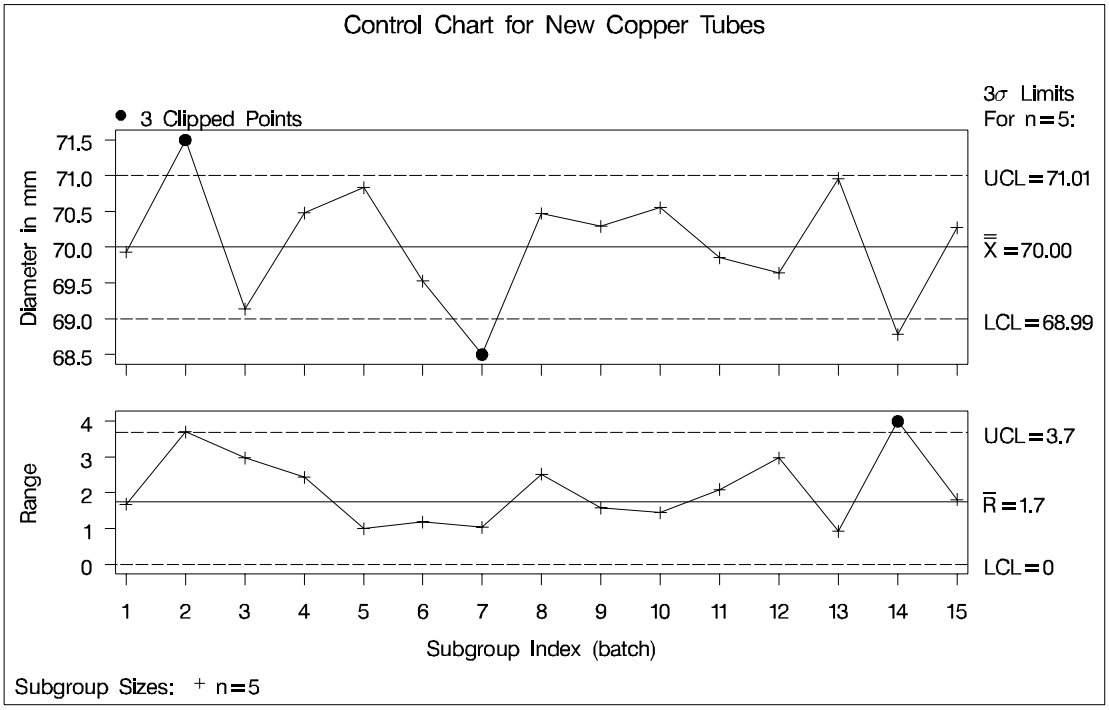


Figure 54.29. $\bar{X}$ and R Charts Using Clipping Options

Specifying CLIPSYMBOL=DOT marks the clipped points with a dot instead of the default square. Specifying CLIPLEGPOS=TOP positions the clipping legend at the top of the chart. The options CLIPLEGEND='# Clipped Points' and CLIPSUBCHAR='#' request the clipping legend *3 Clipped Points*. For more information about the clipping options, see the appropriate entries in Chapter 53, "Dictionary of Options."

Labeling Axes

See SHWLAB in the SAS/QC Sample Library

The SHEWHART procedure provides default labels for the horizontal and vertical axes of control charts. You can specify axis labels by assigning labels to variables, as discussed in the following sections.

Default Labels

If a label is not associated with the *subgroup-variable*, the default horizontal axis label is "Subgroup Index (*subgroup-variable*)." The default vertical axis label for a primary chart identifies the chart type and the process variable. The default vertical axis label for a secondary chart identifies the chart type only.

For example, the following statements create $\bar{X}$ and s charts with default labels using the data set PARTS given in "Displaying Stratified Process Data" on page 1929. The resulting charts are displayed in Figure 54.30.

```
symbol v=none w=1;
title 'Control Chart for Diameter';
proc shewhart history=parts;
   xschart diam*sample;
run;
```

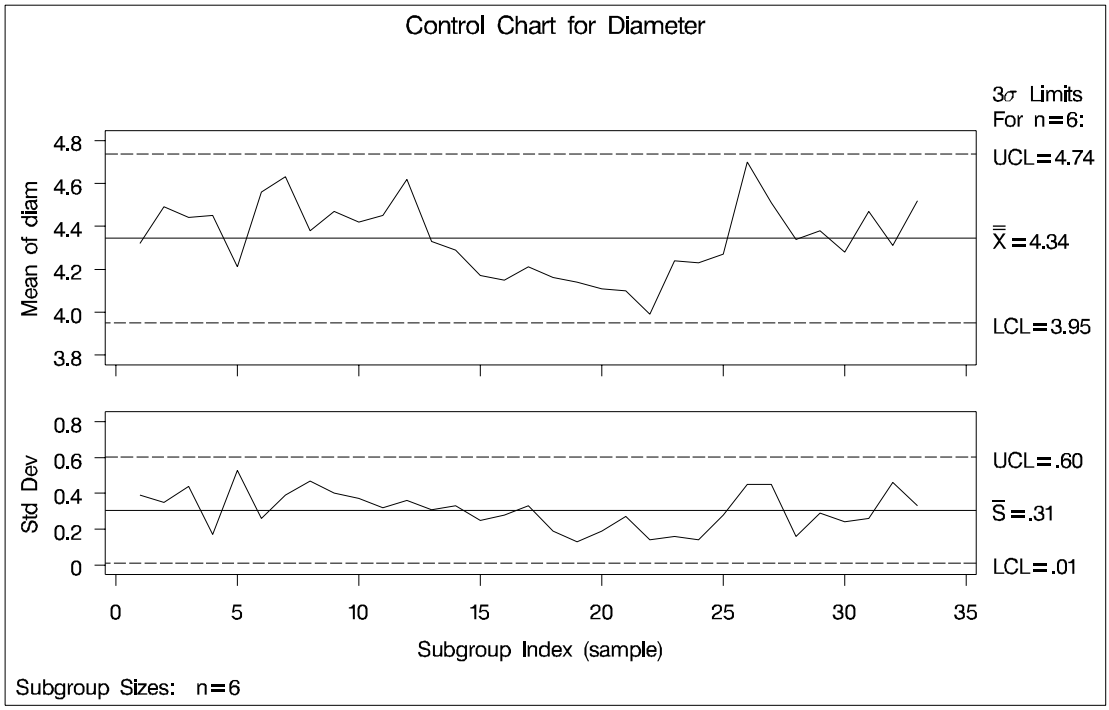


Figure 54.30. Control Charts with Default Labels

Labeling the Horizontal Axis

You can specify a label of up to 40 characters for the horizontal axis by assigning the label to the *subgroup variable* with a LABEL statement (refer to *SAS Language Reference: Dictionary* for a description of LABEL statements). If you use a LABEL statement after the PROC SHEWHART statement and before the RUN statement, the label is associated with the variable only for the duration of the PROC step.

For an example, see page 1969, where Figure 54.31 redisplays the $\bar{X}$ and s charts in Figure 54.30 with specified horizontal and vertical axis labels.

Labeling the Vertical Axis

You can specify a label for the vertical axis of a primary chart by using a LABEL statement to assign the label to a particular variable in the input data set. The type of input data set, the chart statement, and the *process* specified in the chart statement determine which variable to use in the LABEL statement.

- If the input data set is a DATA= data set, assign the label to the process variable (*process*) specified in the chart statement.

- If the input data set is a HISTORY= data set, assign the label to the variable specified in the chart statement whose name begins with the prefix *process* and ends with the appropriate suffix given by the following list:

Chart Statement	Suffix
BOXCHART with CONTROLSTAT=MEAN	X
BOXCHART with CONTROLSTAT=MEDIAN	M
CCHART	U
IRCHART	none
MCHART	M
MRCHART	M
NPCHART	P
PCHART	P
RCHART	R
SCHART	S
UCHART	U
XCHART	X
XRCHART	X
XSCHART	X

  If the prefix *process* consists of 32 characters, shorten the prefix to its first 16 characters and last 15 characters before adding the suffix.

- If the input data set is a TABLE= data set, assign the label to the predefined variable given by the following table:

Chart Statement	Variable
BOXCHART with CONTROLSTAT=MEAN	_SUBX_
BOXCHART with CONTROLSTAT=MEDIAN	_SUBMED_
CCHART	_SUBC_
IRCHART	_SUBI_
MCHART	_SUBMED_
MRCHART	_SUBMED_
NPCHART	_SUBNP_
PCHART	_SUBP_
RCHART	_SUBR_
SCHART	_SUBS_
UCHART	_SUBU_
XCHART	_SUBX_
XRCHART	_SUBX_
XSCHART	_SUBX_

If the chart statement produces primary and secondary charts, as in the case of the XSCHART statement, you can break the label into two parts by including a split character in the label. The part before the split character labels the vertical axis of the primary chart, and the part after the split character labels the vertical axis of the secondary chart. To specify the split character, use the SPLIT= option in the chart statement.

For example, the following statements redisplay the $\bar{X}$ and s charts in Figure 54.30 with specified labels for the horizontal and vertical axes:

```
symbol v=none w=1;
title 'Control Chart for Diameter';
proc shewhart history=parts;
   xschart diam*sample / split   = '/';
   label sample = 'Sample Number'
         diamx  = 'Average Diameter/Std Deviation';
run;
```

The charts are displayed in Figure 54.31. Because the input data set PARTS is a HISTORY= data set, the vertical axes are labeled by assigning a label to the subgroup mean variable DIAMX (that is, the *process* DIAM with the suffix X).* Assigning a label to DIAM would result in an error message since DIAM is interpreted as a prefix rather than a SAS variable.

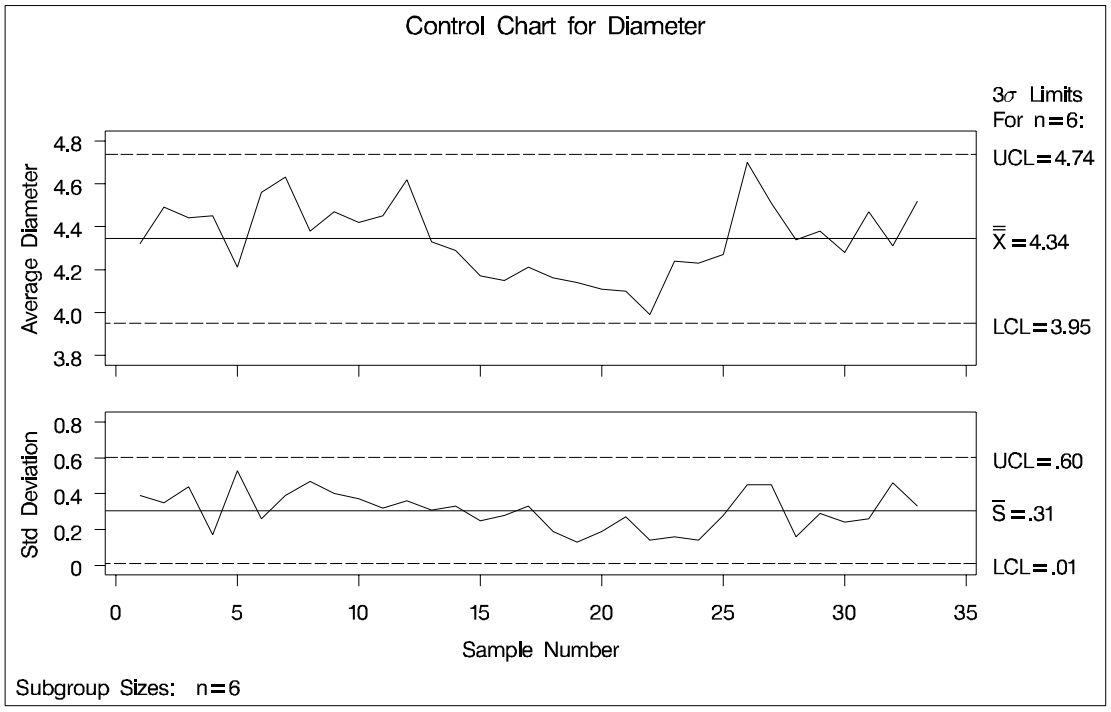


Figure 54.31. Control Charts with Axis Labels Specified

*If the *process* were DIAMETER rather than DIAM, the label would be assigned to the variable DIAMTERX.

If the input data set were a DATA= data set rather than a HISTORY= data set, you would associate the label with the variable DIAM. If the input data set were a TABLE= data set, you would associate the label with the variable _SUBX_.

For another illustration, see Example 43.2 on page 1478.

Selecting Subgroups for Computation and Display

This section describes methods for specifying which subgroups of observations in an input data set (DATA=, HISTORY=, or TABLE=) are to be used to compute control limits and which subgroups are to be displayed as points on the chart.

Using WHERE Statements

See SHWWHR in the SAS/QC Sample Library

The following statements create a data set named BOTTLES that records the number of cracked bottles encountered each day during two months (January and February) of a soft drink bottling operation:

```
data bottles;
informat day date7.;
format day date7.  ;
   nbottles = 3000;
   input day ncracks @@;
datalines;
04JAN94  61  05JAN94  56  06JAN94  71  07JAN94  56
10JAN94  51  11JAN94  64  12JAN94  71  13JAN94  91
14JAN94  98  17JAN94  68  18JAN94  63  19JAN94  60
20JAN94  58  21JAN94  55  24JAN94  78  25JAN94  47
26JAN94  54  27JAN94  69  28JAN94  73  31JAN94  66
01FEB94  57  02FEB94  55  03FEB94  63  04FEB94  50
07FEB94  69  08FEB94  54  09FEB94  64  10FEB94  66
11FEB94  70  14FEB94  49  15FEB94  57  16FEB94  56
17FEB94  59  18FEB94  66  21FEB94  60  22FEB94  58
23FEB94  67  24FEB94  60  25FEB94  62  28FEB94  48
;
run;
```

The variable NBOTTLES contains the number of bottles sampled each day, and the variable NCRACKS contains the number of cracked bottles in each sample.

The following statements create a p chart for the number of cracked bottles based on the January production:

```
symbol h=3.0 pct;
title 'Preliminary Analysis of January Production';
proc shewhart data=bottles;
   where day <= '31JAN94'D;
   pchart ncracks * day / subgroupn = nbottles
                          nohlabel
                          nolegend
                          outlimits = mylim;
   label ncracks = 'Proportion With Cracks';
run;
```

The chart is shown in Figure 54.32. The WHERE statement restricts the observations read from BOTTLES so that the control limits are estimated from the January data, and only the January data are displayed on the chart. For details concerning the WHERE statement, refer to *SAS Language Reference: Dictionary*.

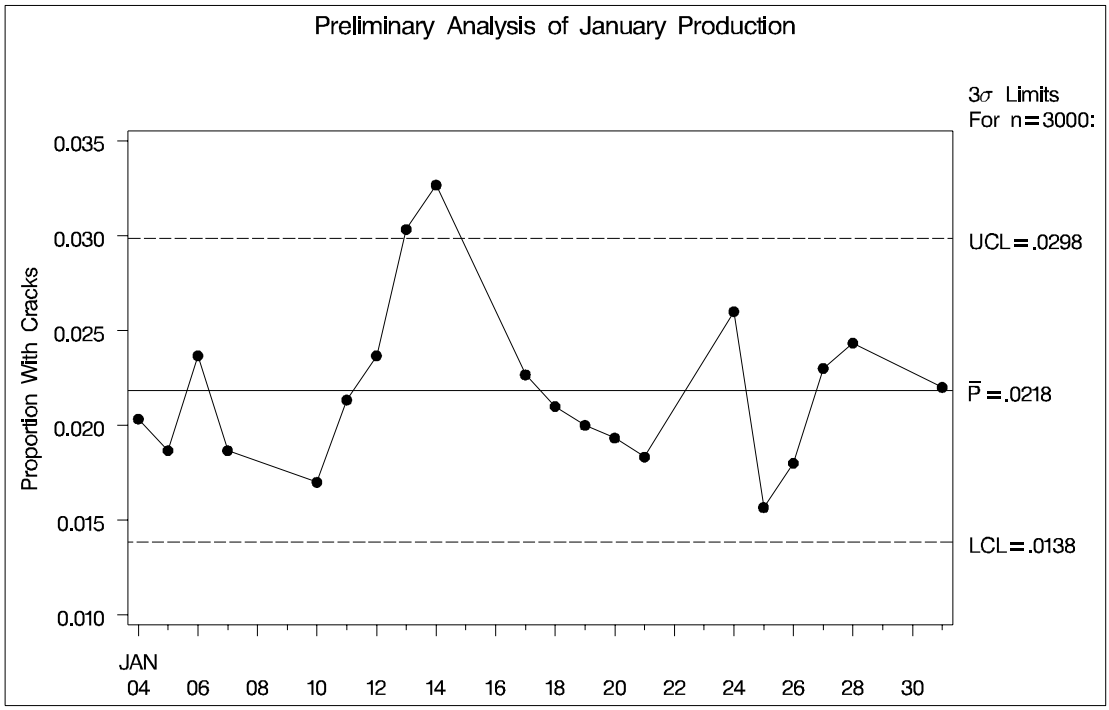


Figure 54.32. Preliminary p Chart for January Data

In Figure 54.32, a special cause of variation is signaled by the proportions for January 13 and January 14, which exceed the upper control limit. Since the cause, an improper machine setting, was corrected, it is appropriate to recompute the control limits by excluding the data for these two days. Again, this can be done with a WHERE statement, as follows:

```
title 'Final Analysis of January Production';
proc shewhart data=bottles;
   where ( day <= '31JAN94'D ) &
         ( day ne '13JAN94'D ) &
         ( day ne '14JAN94'D )   ;
   pchart ncracks * day / subgroupn = nbottles
                          nohlabel
                          nolegend
                          outlimits = janlim;
   label ncracks = 'Proportion With Cracks';
run;
```

The chart is shown in Figure 54.33.

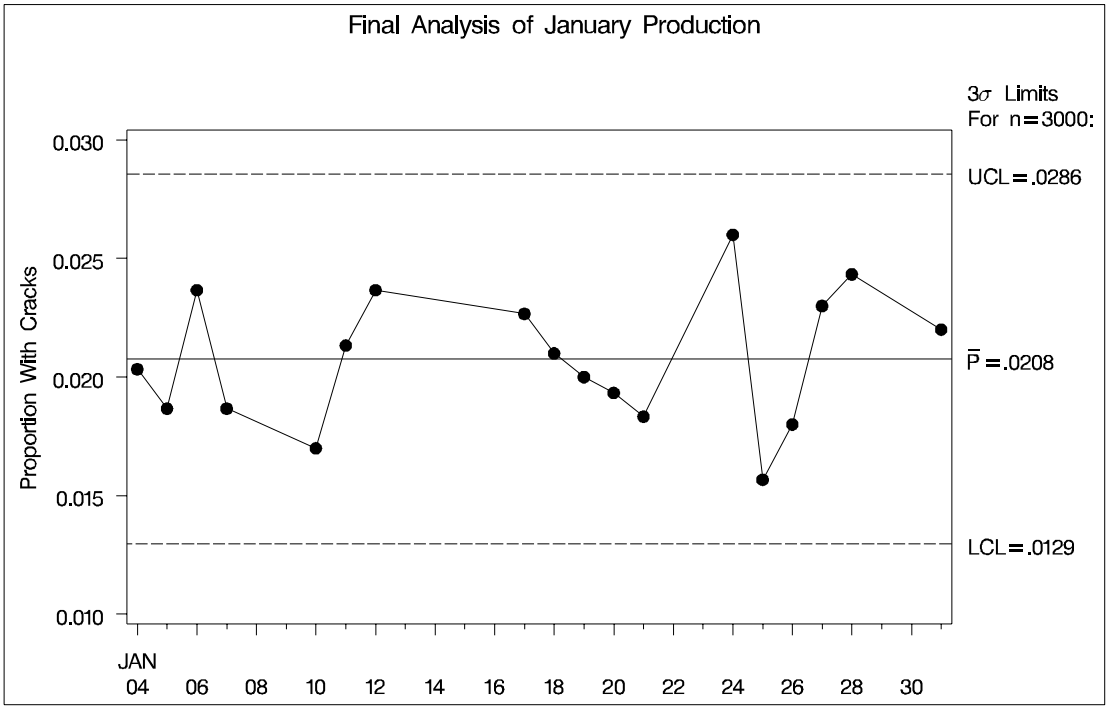


Figure 54.33. Final p Chart for January Data

The data set JANLIM, which saves the control limits, is listed in Figure 54.34.

VARP	_SUBGP_	_TYPE_	_LIMIT_	_ALPHA_	_SIGMAS_	_LCLP_	_P_	_UCLP_
ncracks	day	ESTIMATE	3000	.003072976	3	0.012950	0.020759	0.028569

Figure 54.34. Listing of the LIMITS= Data Set JANLIM

Now, the control limits based on the January data are to be applied to the February data. Again, this can be done with a WHERE statement, as follows:*

```
title 'Analysis of February Production';
proc shewhart data=bottles limits=janlim;
   where day > '31JAN94'D;
   pchart ncracks * day / subgroupn = nbottles
                          nolegend
                          nohlabel;
   label ncracks = 'Proportion With Cracks';
run;
```

*In Release 6.09 and in earlier releases, it is also necessary to specify the READLIMITS option to read control limits from a LIMITS= data set.

The chart is shown in Figure 54.35.

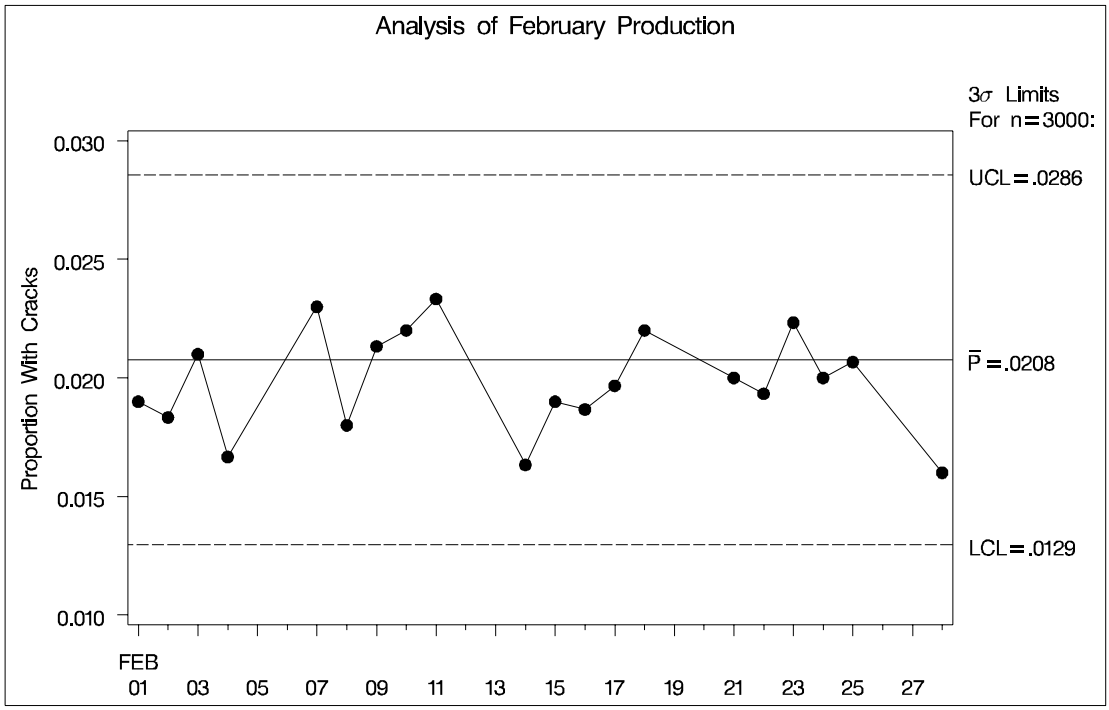


Figure 54.35. p Chart for February Data

Using Switch Variables

As an alternative to reading a LIMITS= data set and using a WHERE statement, you can provide two special switch variables named _COMP_ and _DISP_ in the input data set. The rules for using these variables are as follows:

See SHWSVAR in the SAS/QC Sample Library

- Switch variables must be character variables of length one. Valid values for these variables are `Y` (or `y`) and `N` (or `n`). A blank value is treated as `Y`.
- Subgroups for which _COMP_ is equal to `Y` are included in computations of parameter estimates and control limits, and observations for which _COMP_ is equal to `N` are excluded.
- Subgroups for which _DISP_ is equal to `Y` are displayed on the chart, and subgroups for which _DISP_ is equal to `N` are not displayed.
- If the chart statement creates a chart for variables, you can provide two additional switch variables named _COMP2_ and _DISP2_, which are defined similarly to _COMP_ and _DISP_. In this case, the variable _COMP_ specifies which subgroups are used to estimate the process mean μ, and the variable _COMP2_ specifies which subgroups are used to estimate the process standard deviation σ. The variable _DISP_ specifies which subgroups are displayed on the primary chart ($\bar{X}$ chart, median chart, or individual measurements chart), and the variable _DISP2_ specifies which subgroups are displayed on the secondary chart (R chart or s chart).

- The variables _COMP_ and _COMP2_ are not applicable when control limits or control limit parameters are read from a LIMITS= data set.
- The variables _DISP_ and _DISP2_ take precedence over the display controlled by the LIMITN= and ALLN options.
- If the input data set is a DATA= data set with multiple observations per subgroup, switch variable values must be constant within a subgroup.
- Switch variables are saved in OUTHISTORY= and OUTTABLE= data sets. Subgroups for which _DISP_ is equal to N are not saved in an OUTTABLE= data set, and such subgroups are not displayed in tables created with the TABLE and related options.

The following statements illustrate how the switch variables _COMP_ and _DISP_ can be used with the bottle production data:

```
data bottles;
   length _comp_ _disp_ $ 1;
   set bottles;
   if      day  = '13JAN94'D then _comp_ = 'n';
   else if day  = '14JAN94'D then _comp_ = 'n';
   else if day <= '31JAN94'D then _comp_ = 'y';
   else                           _comp_ = 'n';
   if      day <= '31JAN94'D then _disp_ = 'n';
   else                           _disp_ = 'y';
run;

title 'Analysis of February Production';
proc shewhart data=bottles;
   pchart ncracks * day / subgroupn = nbottles
                          nolegend
                          nohlabel;
   label ncracks = 'Proportion With Cracks';
run;
```

The chart is identical to the chart in Figure 54.35.

In general, switch variables are more versatile than WHERE statements in applications where subgroups are simultaneously selected for computation and display. Switch variables also provide a permanent record of which subgroups were selected. The WHERE statement does not alter the input data set; it simply restricts the observations that are read; consequently, the WHERE statement can be more efficient than switch variables for processing large data sets.

Chapter 55
Tests for Special Causes

Chapter Contents

STANDARD TESTS FOR SPECIAL CAUSES 1977
- Requesting Standard Tests 1979
- Interpreting Standard Tests for Special Causes 1981
- Modifying Standard Tests for Special Causes 1982
- Applying Tests with Varying Subgroup Sample Sizes 1983
- Labeling Signaled Points with a Variable 1985
- Applying Tests with Multiple Phases 1986
- Applying Tests with Multiple Sets of Control Limits 1987
- Enhancing the Display of Signaled Tests 1990

NONSTANDARD TESTS FOR SPECIAL CAUSES 1991
- Applying Tests to Range and Standard Deviation Charts 1991
- Applying Tests Based on Generalized Patterns 1992
- Customizing Tests with DATA Step Programs 1995

Chapter 55
Tests for Special Causes

This chapter provides details concerning standard and nonstandard tests for special causes that you can apply with the SHEWHART procedure.

Standard Tests for Special Causes

The SHEWHART procedure provides eight standard *tests for special causes*, also referred to as *rules for lack of control*, *supplementary rules*, *runs tests*, *runs rules*, *pattern tests*, and *Western Electric rules*. These tests improve the sensitivity of the Shewhart chart to small changes in the process. * You can also improve the sensitivity of the chart by increasing the rate of sampling, increasing the subgroup sample size, and using control limits that represent less than three standard errors of variation from the central line. However, increasing the sampling rate and sample size is often impractical, and tightening the control limits increases the chances of falsely signaling an out-of-control condition. By detecting particular nonrandom patterns in the points plotted on the chart, the tests can provide greater sensitivity and useful diagnostic information while incurring a reasonable probability of a false signal.

The patterns detected by the eight standard tests are defined in Table 55.1 and Table 55.2, and they are illustrated in Figure 55.1 and Figure 55.2. All eight tests were developed for use with fixed 3σ limits. The tests are indexed according to the numbering sequence used by Nelson (1984, 1985). You can request any combination of the eight tests by specifying the test *indexes* with the TESTS= option in the BOXCHART, CCHART, IRCHART, MCHART, MRCHART, NPCHART, PCHART, UCHART, XCHART, XRCHART, and XSCHART statements.

The following restrictions apply to the tests:

- Only Tests 1, 2, 3, and 4 are recommended for c charts, np charts, p charts, and u charts created with the CCHART, NPCHART, PCHART, and UCHART statements, respectively. In these four cases, Test 2 should not be used unless the process distribution is symmetric or nearly symmetric.
- By default, the TESTS= option is not applied with control limits that are not 3σ limits or that vary with subgroup sample size. You can use the NO3SIGMACHECK option to request tests for special causes when the SIGMAS= option specifies control limits other than 3σ limits. This is not recommended for standard control chart applications, since the standard tests for special causes are based on 3σ limits. You can apply test for special causes when control limits vary with subgroup sample size by using the LIMITN= or TESTNMETHOD= options (see page 1980 and page 1983).

*Cumulative sum control charts and moving average control charts also detect small shifts more quickly than an ordinary Shewhart chart. See Chapter 18, "PROC CUSUM Statement," and Chapter 26, "PROC MACONTROL Statement," for more information.

Table 55.1. Definitions of Tests 1 to 4

Test *Index*	Pattern Description
1	One point beyond Zone A (outside the control limits)
2	Nine points in a row in Zone C or beyond on one side of the central line (see Note 1 below)
3	Six points in a row steadily increasing or steadily decreasing (see Note 2 below)
4	Fourteen points in a row alternating up and down

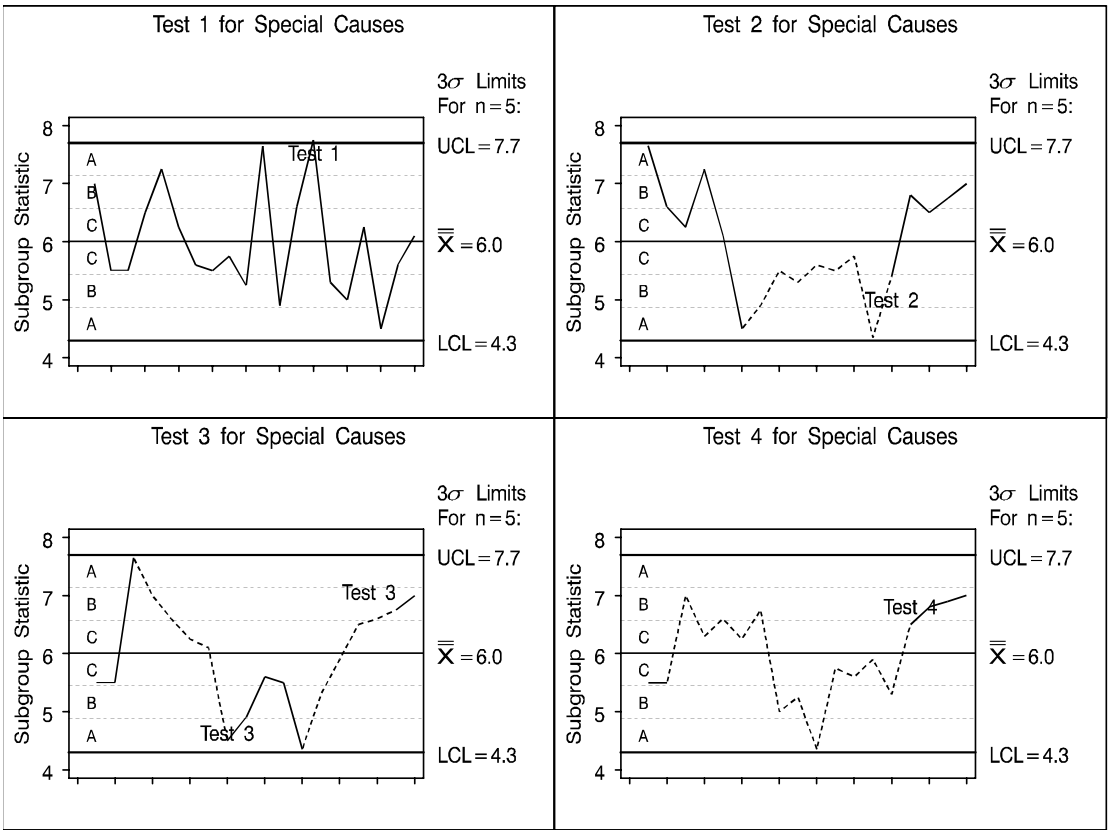


Figure 55.1. Examples of Tests 1 to 4

Notes:

1. The number of points in Test 2 can be specified as 7, 8, 9, 11, 14, or 20 with the TEST2RUN= option.
2. The number of points in Test 3 can be specified as 6, 7, or 8 with the TEST3RUN= option.

Table 55.2. Definitions of Tests 5 to 8

Test *Index*	Pattern Description
5	Two out of three points in a row in Zone A or beyond
6	Four out of five points in a row in Zone B or beyond
7	Fifteen points in a row in Zone C on either or both sides of the central line
8	Eight points in a row on either or both sides of the central line with no points in Zone C

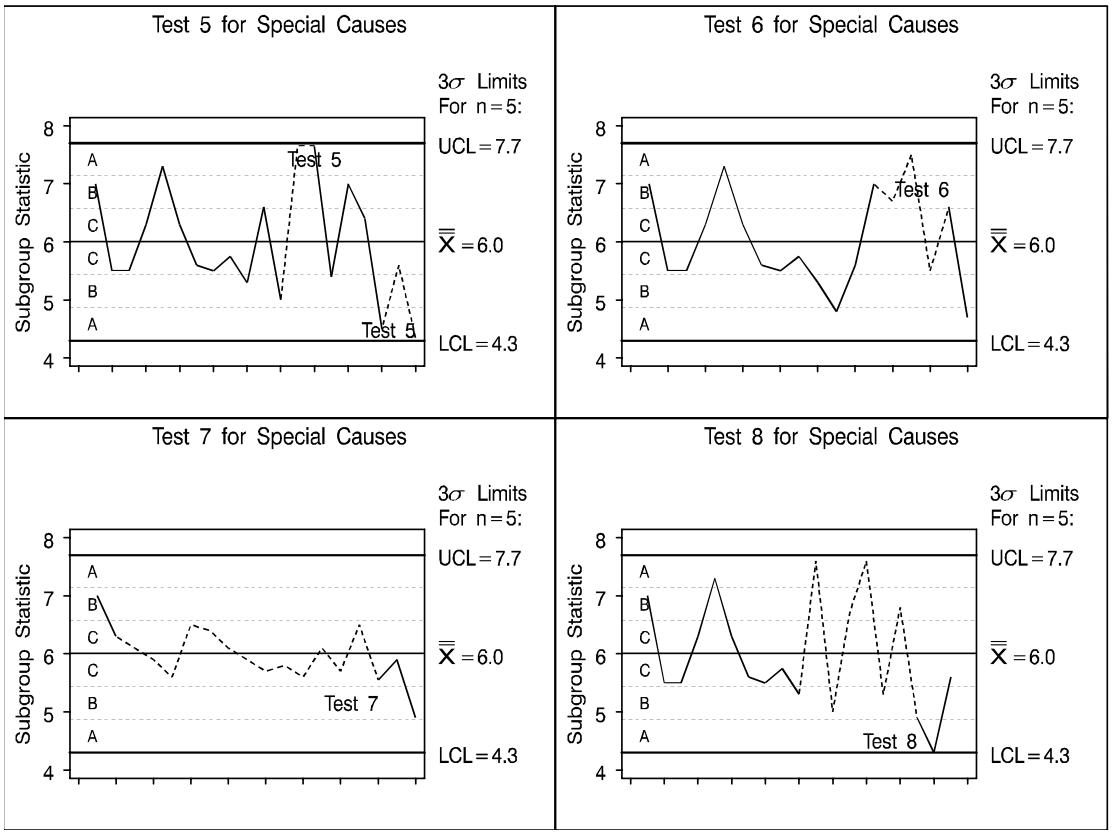


Figure 55.2. Examples of Tests 5 to 8

Requesting Standard Tests

The following example illustrates how to request the standard tests for special causes. The tests are applied to an $\bar{X}$ chart for assembly offset measurements whose subgroup means, ranges, and sample sizes are provided by the variables OFFSETX, OFFSETR, and OFFSETN, respectively, in a data set named ASSEMBLY.*

See SHWTSC1 in the SAS/QC Sample Library

*The data set ASSEMBLY is also used by subsequent examples in this chapter.

```
data assembly;
   length system $ 1 comment $ 16;
   label sample = 'Sample Number';
   input system sample offsetx offsetr offsetn comment $16. ;
datalines;
T   1  19.80  3.8  5
T   2  17.16  8.3  5
T   3  20.11  6.7  5
T   4  20.89  5.5  5
T   5  20.83  2.3  5
T   6  18.87  2.6  5
T   7  20.84  2.3  5
T   8  23.33  5.7  5  New Tool
T   9  19.21  3.5  5
T  10  20.48  3.2  5
T  11  22.05  4.7  5
T  12  20.02  6.7  5
T  13  17.58  2.0  5
T  14  19.11  5.7  5
T  15  20.03  4.1  5
R  16  20.56  3.7  5  Changed System
R  17  20.86  3.3  5
R  18  21.10  5.6  5  Reset Tool
R  19  19.05  2.7  5
R  20  21.76  2.8  5
R  21  21.76  6.4  5
R  22  20.54  4.8  5
R  23  20.04  8.2  5
R  24  19.94  8.8  5
R  25  20.70  5.1  5
Q  26  21.40 12.1  7  Bad Reading
Q  27  21.32  3.2  7
Q  28  20.03  5.2  7  New Gauge
Q  29  22.02  5.9  7
Q  30  21.32  4.3  7
;
run;
```

The following statements use the TESTS= option to request Tests 1 to 4. Note that the *process* OFFSET is specified in the XRCHART statement to indicate that the three summary variables OFFSETX, OFFSETR, and OFFSETN are to be read from the HISTORY= data set ASSEMBLY.

```
title 'Analysis of Assembly Data';
proc shewhart history=assembly;
   xrchart offset * sample / mu0      = 20
                             sigma0   = 2.24
                             limitn   = 5
                             alln
                             tests    = 1 to 4
                             ltests   = 2
                             vaxis    = 16 to 26 by 2
                             split    = '/';
   label offsetx = 'Avg Offset in cm/Range';
run;
```

The chart is displayed in Figure 55.3. Test 1 is positive at the 8^{th} subgroup, and Test 3 is positive at the 18^{th} subgroup.

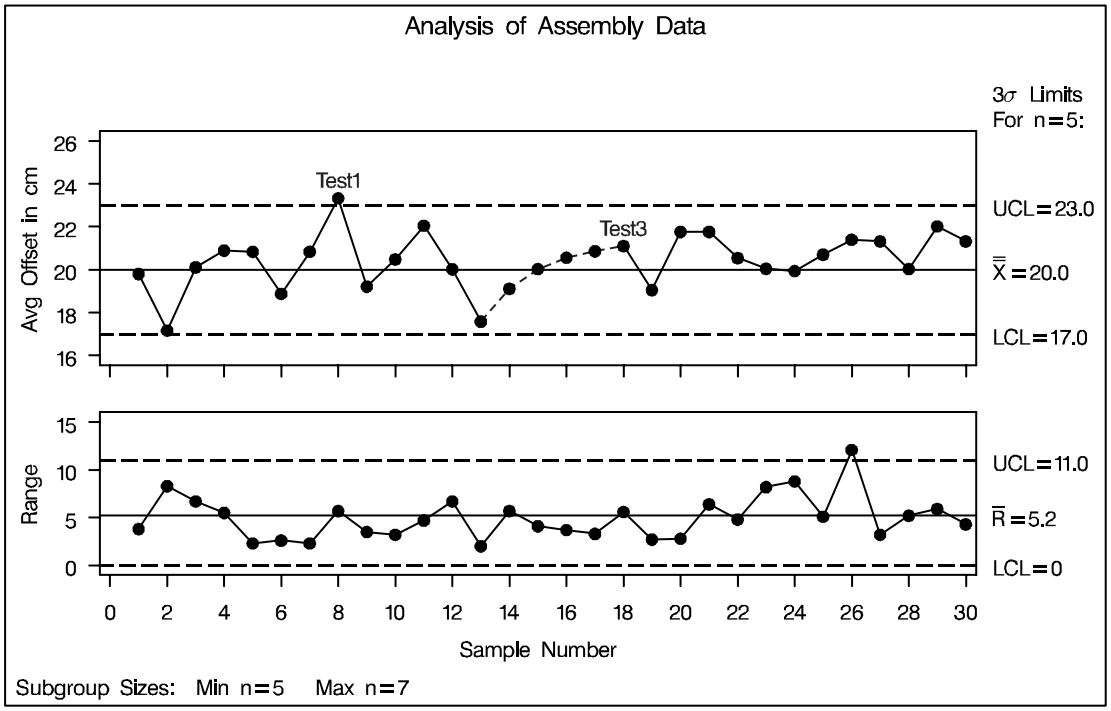


Figure 55.3. Standard Tests Using the TESTS= Option

The control limits in Figure 55.3 are based on standard values for the process mean and standard deviation specified with the MU0= and SIGMA0= options, respectively. Although the subgroup sizes vary, fixed control limits are displayed corresponding to a nominal sample size of five, which is specified with the LIMITN= option. Since ALLN is specified, points are displayed for all subgroups, regardless of sample size.

Note: If the LIMITN= option were not specified, the control limits would vary with subgroup sample size, and by default the tests would not be applied. An alternative method for applying the tests with varying subgroup sample sizes is discussed in "Applying Tests with Varying Subgroup Sample Sizes" on page 1983.

Interpreting Standard Tests for Special Causes

Nelson (1984, 1985) makes the following comments concerning the interpretation of the tests:

- When a process is in statistical control, the chance of a false signal for each test is less than five in one thousand.
- Test 1 is positive if there is a shift in the process mean, if there is an increase in the process standard deviation, or if there is a "single aberration in the process such as a mistake in calculation, an error in measurement, bad raw material, a breakdown of equipment, and so on" (Nelson 1985).

- Test 2 signals a shift in the process mean. The use of nine points (rather than seven as in Grant and Leavenworth 1988) for the pattern that defines Test 2 makes the chance of a false signal comparable to that of Test 1. (To control the number of points for the pattern in test 2, use the TEST2RUN= option in the chart statement.)
- Test 3 signals a drift in the process mean. Nelson (1985) states that causes can include "tool wear, depletion of chemical baths, deteriorating maintenance, improvement in skill, and so on."
- Test 4 signals "a systematic effect such as produced by two machines, spindles, operators or vendors used alternately" (Nelson 1985).
- Tests 1, 2, 3, and 4 should be applied routinely; the combined chance of a false signal from one or more of these tests is less than one in a hundred. Nelson (1985) describes these tests as "a good set that will react to many commonly occurring special causes."
- In the case of charts for variables, the first four tests should be augmented by Tests 5 and 6 when earlier warning is desired. The chance of a false signal increases to two in a hundred.
- Tests 7 and 8 indicate stratification (observations in a subgroup have multiple sources with different means). Test 7 is positive when the observations in the subgroup always have multiple sources. Test 8 is positive when the subgroups are taken from one source at a time.

Nelson (1985) also comments that "the probabilities quoted for getting false signals should not be considered to be very accurate" since the probabilities are based on assumptions of normality and independence that may not be satisfied. Consequently, he recommends that the tests "should be viewed as simply practical rules for action rather than tests having specific probabilities associated with them." Nelson cautions that "it is possible, though unlikely, for a process to be out of control yet not show any signals from these eight tests."

Modifying Standard Tests for Special Causes

Some textbooks and references present slightly different versions of Tests 2 and 3. You can use the following options to request these modifications:

- TEST2RUN=*run-length* specifies the length of the pattern for Test 2. The form of the test for each *run-length* is given in the following table. The default *run-length* is 9.

Run-length	Number of Points on One Side of Central Line
7	7 in a row
8	8 in a row
9	9 in a row
11	at least 10 out of 11 in a row
14	at least 12 out of 14 in a row
20	at least 16 out of 20 in a row

- TEST3RUN=*run-length* specifies the length of the pattern for Test 3. The *run-length* values allowed are 6, 7, and 8. The default *run-length* is 6.

The Western Electric Company (now AT&T) *Statistical Quality Control Handbook* and Montgomery (1996) discuss a test that is signaled by eight points in a row in Zone C or beyond (on one side of the central line). You can request this test by specifying TESTS=2 and TEST2RUN=8. The *Handbook* also discusses tests corresponding to Tests 1, 5, 6, 7, and 8.

Kume (1985) recommends a number of tests for special causes that can be regarded as modifications of Tests 2 and 3:

- seven points in a row on one side of the central line. Specify TESTS=2 and TEST2RUN=7.
- at least 10 out of 11 points in a row on one side of the central line. Specify TESTS=2 and TEST2RUN=11.
- at least 12 out of 14 points in a row on one side of the central line. Specify TESTS=2 and TEST2RUN=14.
- at least 16 out of 20 points in a row on one side of the central line. Specify TESTS=2 and TEST2RUN=20.
- seven points in a row steadily increasing or decreasing. Specify TESTS=3 and TEST3RUN=7.

Applying Tests with Varying Subgroup Sample Sizes

See SHWTSC1 in the SAS/QC Sample Library

Nelson (1989, 1994) describes the use of standardization to apply the tests for special causes to data involving varying subgroup samples. This approach applies the tests to the standardized subgroup statistics, setting the control limits at ± 3 and the zone boundaries at ± 1 and ± 2. For instance, for an $\bar{X}$ chart with subgroup means $\bar{X}_i$ and varying subgroup sample sizes n_i, the tests are applied to the standardized values $z_i = (\bar{X}_i - \overline{\overline{X}})/(s/\sqrt{n_i})$, where $\overline{\overline{X}}$ estimates the process mean, and s estimates the process standard deviation. You can request this method with the TESTNMETHOD= option,* as illustrated by the following statements:

```
title 'Analysis of Assembly Data';
proc shewhart history=assembly;
   xrchart offset * sample / mu0         = 20
                             sigma0      = 2.24
                             tests       = 1 to 4
                             testnmethod = standardize
                             testlabel   = space
                             ltests      = 2
                             vaxis       = 16 to 26 by 2
                             split       = '/';
   label offsetx = 'Avg Offset in cm/Range';
run;
```

*If the TESTNMETHOD= option were omitted in this example, the tests would not be applied, and a warning message would be displayed in the SAS log.

Here the tests are applied to $z_i = (\bar{X}_i - 20)/(2.24/\sqrt{n_i})$. The chart, shown in Figure 55.4, displays the results of the tests on a plot of the *unstandardized* means.

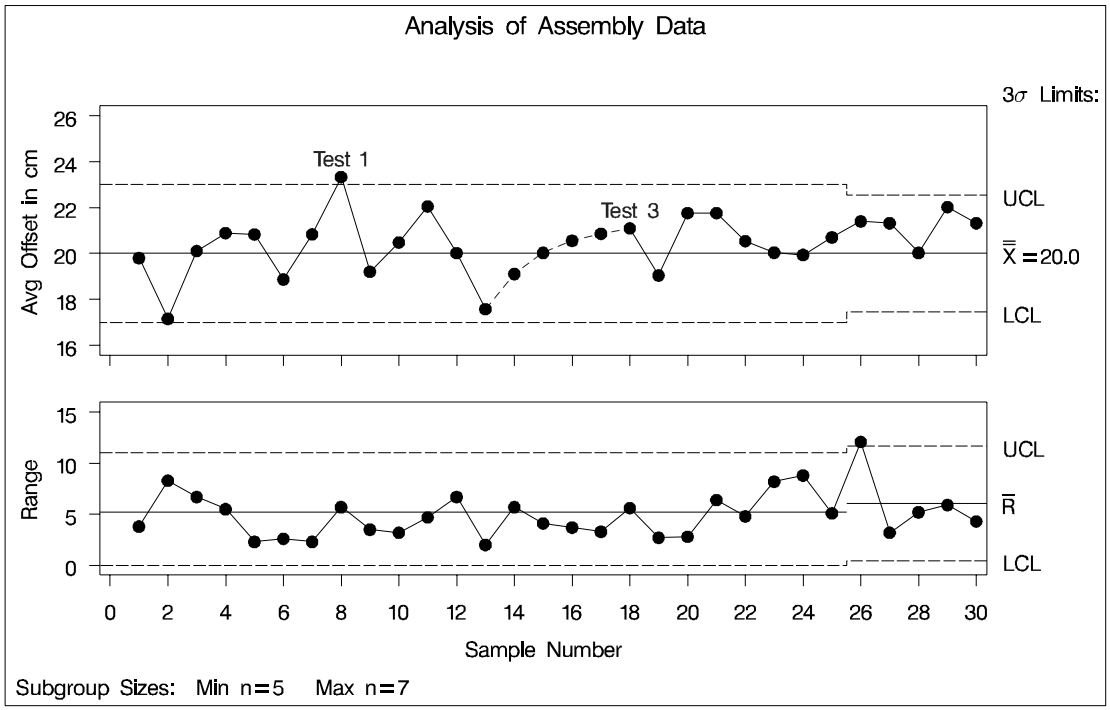


Figure 55.4. The TESTNMETHOD= Option for Varying Subgroup Sizes

The following statements create an equivalent chart that plots the *standardized* means:

```
data assembly;
   set assembly;
   zx = ( offsetx - 20 ) / ( 2.24 / sqrt( offsetn ) );
run;

title 'Analysis of Standardized Assembly Data';
proc shewhart
   history=assembly (rename = (offsetr=zr offsetn=zn));
   xrchart z * sample / mu0          = 0
                        sigma0       = 2.2361 /* sqrt 5 */
                        limitn       = 5
                        alln
                        tests        = 1 to 4
                        testlabel    = space
                        ltests       = 2
                        vaxis        = -4 to 6 by 2
                        split = '/';
   label zx = 'Std Avg Offset/Range';
run;
```

Here, the SIGMA0= value is the square root of the LIMITN= value. The chart is shown in Figure 55.5.

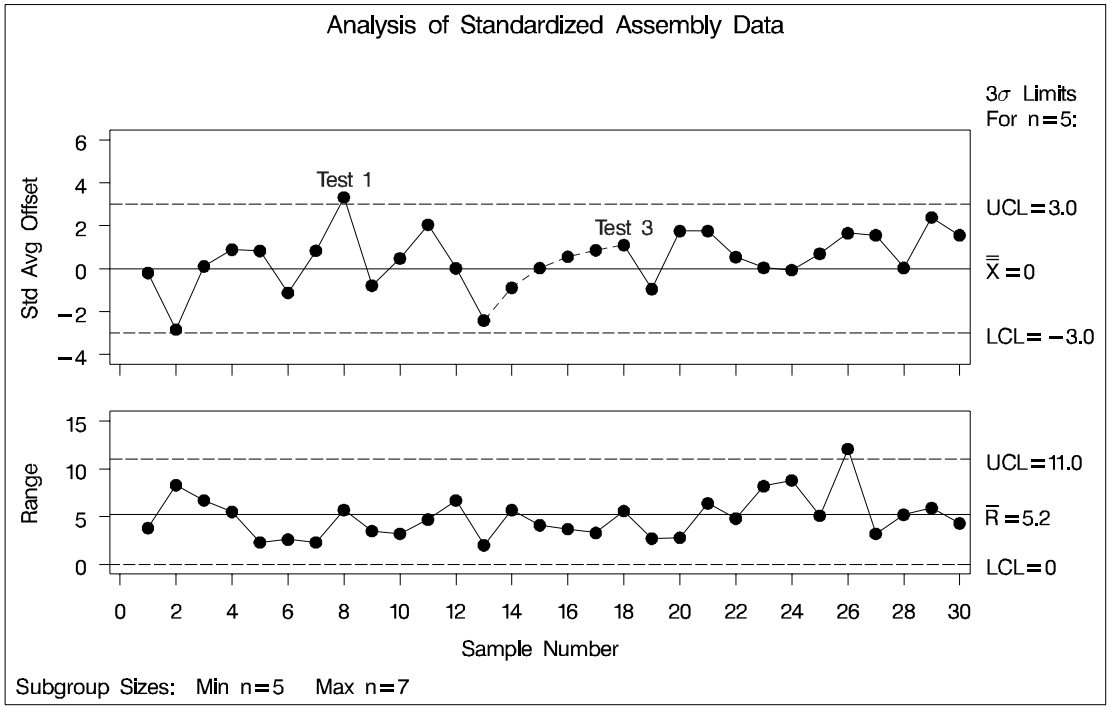


Figure 55.5. Tests with Standardized Means

Note: In situations where the standard deviation is estimated from the data and the subgroup sample sizes vary, you can use the SMETHOD= option to request various estimation methods, including the method of Burr (1969).

Labeling Signaled Points with a Variable

If a test is signaled at a particular point, the point is labeled by default with the *index* of the test, as illustrated in Figure 55.3.* You can use the TESTLABEL= option to specify a variable in the input data set whose *values* provide the labels, as illustrated by the following statements:

See SHWTSC1 in the SAS/QC Sample Library

```
title 'Analysis of Assembly Data';
proc shewhart history=assembly;
   xrchart offset * sample / mu0        = 20
                             sigma0     = 2.24
                             limitn     = 5
                             alln
                             tests      = 1 to 4
                             testlabel  = ( comment )
                             ltests     = 2
                             vaxis      = 16 to 24 by 2
                             split = '/';
   label offsetx = 'Avg Offset in cm/Range';
run;
```

*If two or more tests are positive at a particular point, the default label identifies the *index* of the test that was specified first with the TESTS= option.

The labels are shown in Figure 55.6. It is often helpful to specify a variable with the TESTLABEL= option that provides operator comments or other information that can aid in the identification of special causes.

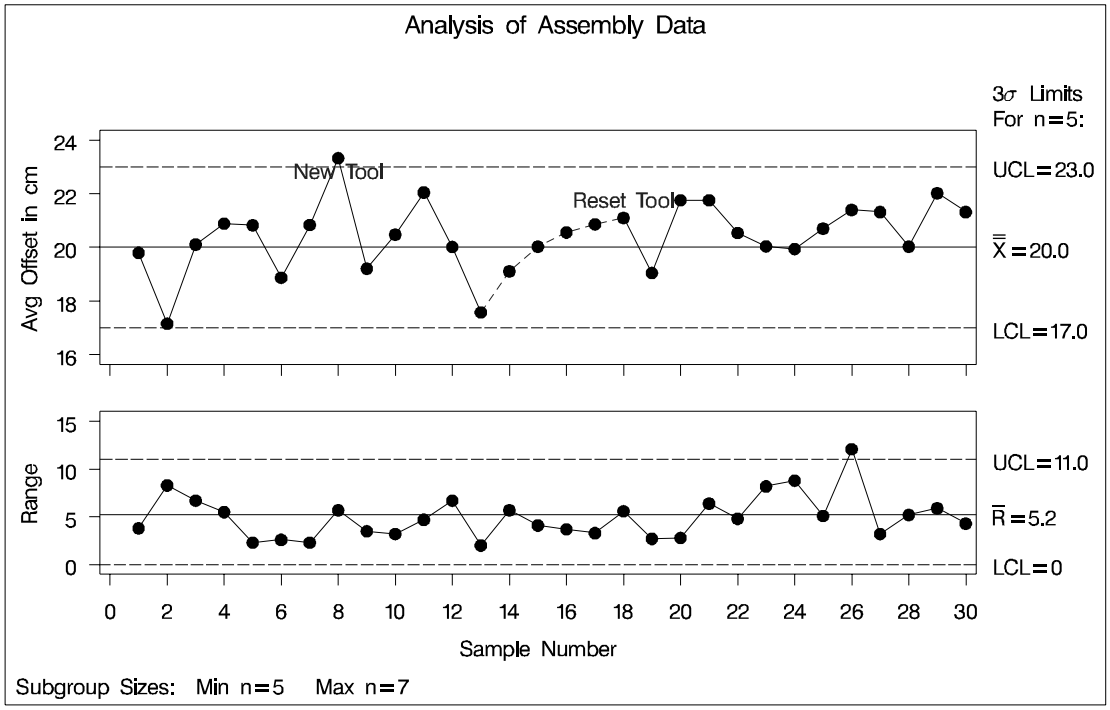


Figure 55.6. Labeling Points with a TESTLABEL= Variable

Applying Tests with Multiple Phases

See SHWTSC1 in the SAS/QC Sample Library

The data set ASSEMBLY includes a variable named SYSTEM, which indicates the manufacturing system used to produce each assembly. As shown by the following statements, this variable can be temporarily renamed and read as the variable _PHASE_ to create a control chart that displays the *phases* (groups of consecutive subgroups) for which SYSTEM is equal to T, R, and Q:

```
title 'Manufacturing Systems Used in Assembly';
proc shewhart
   history=assembly (rename=(system=_phase_));
   xrchart offset * sample /
      mu0          = 20
      sigma0       = 2.24
      limitn       = 5
      alln
      tests        = 1 to 4
      testlabel    = ( comment )
      ltests       = 2
      readphases   = ('T' 'R' 'Q')
      phaselegend
      phaseref
      vaxis        = 16 to 26 by 2
      split        = '/';
   label offsetx = 'Avg Offset in cm/Range';
run;
```

The chart is shown in Figure 55.7.

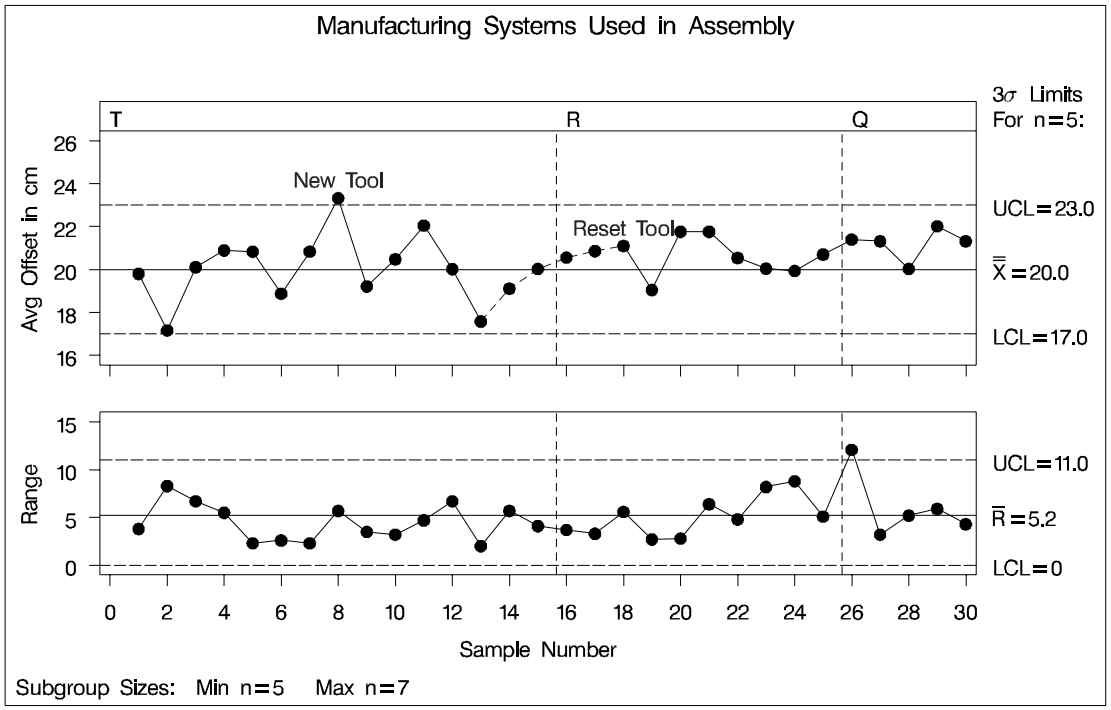


Figure 55.7. Single Set of Limits with Multiple Phases

Note that a single set of fixed 3σ limits is displayed for all three phases because LIMITN=5 and ALLN are specified. Consequently, the tests requested with the TESTS= option are applied independently of the phases. In general, however, it is possible to display distinct sets of control limits for different phases, and in such situations, the tests are not applied independently of phases, as discussed in the next example.

Applying Tests with Multiple Sets of Control Limits

This example is a continuation of the previous example, except that distinct control limits are displayed for each of the phases determined by the variable SYSTEM. The control limit parameters (mean, standard deviation, and nominal sample size) for each phase (manufacturing system) are provided in the following data set:

See SHWTSC2 in the SAS/QC Sample Library

```
data syslim;
   length _var_ $8 _subgrp_ $8 _type_ $8 _index_ $1;
input _var_ _subgrp_ _index_ _type_ _mean_ _stddev_
      _limitn_ _sigmas_;
datalines;
offset sample R standard 20.5 2.02 5 3
offset sample Q standard 20.2 2.35 7 3
offset sample T standard 20.0 2.24 5 3
;
run;
```

The following statements read the control limit parameters from SYSLIM and use the READPHASES= and READINDEXES= options to display a distinct set of control limits for each phase:

```
title 'System-Specific Control Limits';
proc shewhart
   limits=syslim
   history=assembly (rename=(system=_phase_));
   xrchart offset * sample /
      tests       = 1 to 4
      testlabel   = ( comment )
      readindexes = ('T' 'R' 'Q')
      readphases  = ('T' 'R' 'Q')
      phaselegend
      phaseref
      phasebreak
      vaxis       = 16 to 26 by 2
      split       = '/' ;
   label offsetx = 'Avg Offset in cm/Range';
run;
```

The chart is shown in Figure 55.8. The tests requested with the TESTS= option are applied strictly within the phases, since the control limits are not constant across the phases (as in Figure 55.7). In particular, note that the pattern labeled *Reset Tool* in Figure 55.7 is not detected in Figure 55.8.

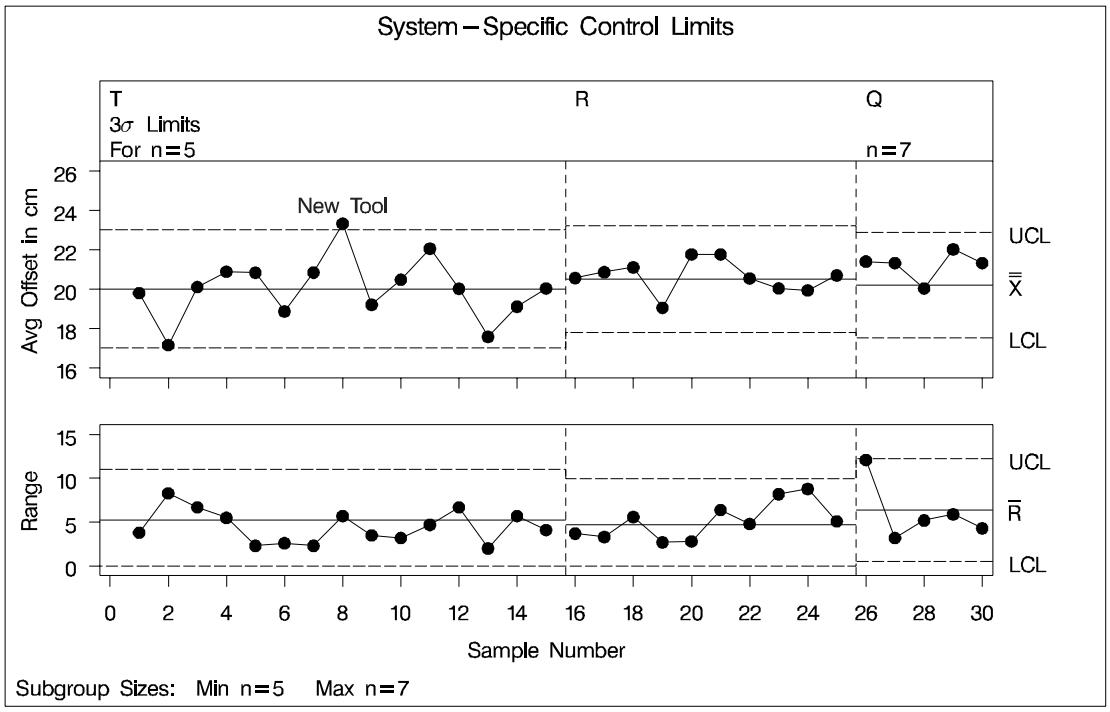


Figure 55.8. Multiple Sets of Control Limits

In most applications involving multiple control limits, a known change or improvement has occurred at the beginning of each phase; consequently, it is appropriate to restart the tests at the beginning of each phase rather than search for patterns that

span the boundaries of consecutive phases. In these situations, the PHASEBREAK option is useful for suppressing the connection of points from one phase to the next. Note that it is not necessary to specify the TESTNMETHOD= option here because the subgroup sample sizes are constant within each phase.

There may be applications in which it is appropriate to apply the tests across phase boundaries. You can use the TESTACROSS option to request this behavior.

```
title 'System-Specific Control Limits';
proc shewhart
   limits=syslim
   history=assembly (rename=(system=_phase_));
   xrchart offset * sample /
      tests       = 1 to 4
      testlabel   = ( comment )
      testnmethod = standardize
      testacross
      ltests      = 2
      readindexes = ('T' 'R' 'Q')
      readphases  = ('T' 'R' 'Q')
      phaselegend
      phaseref
      vaxis       = 16 to 26 by 2
      cinfill     = ligr
      split       = '/';
   label offsetx = 'Avg Offset in cm/Range';
run;
```

The chart created with the TESTACROSS option is displayed in Figure 55.9.

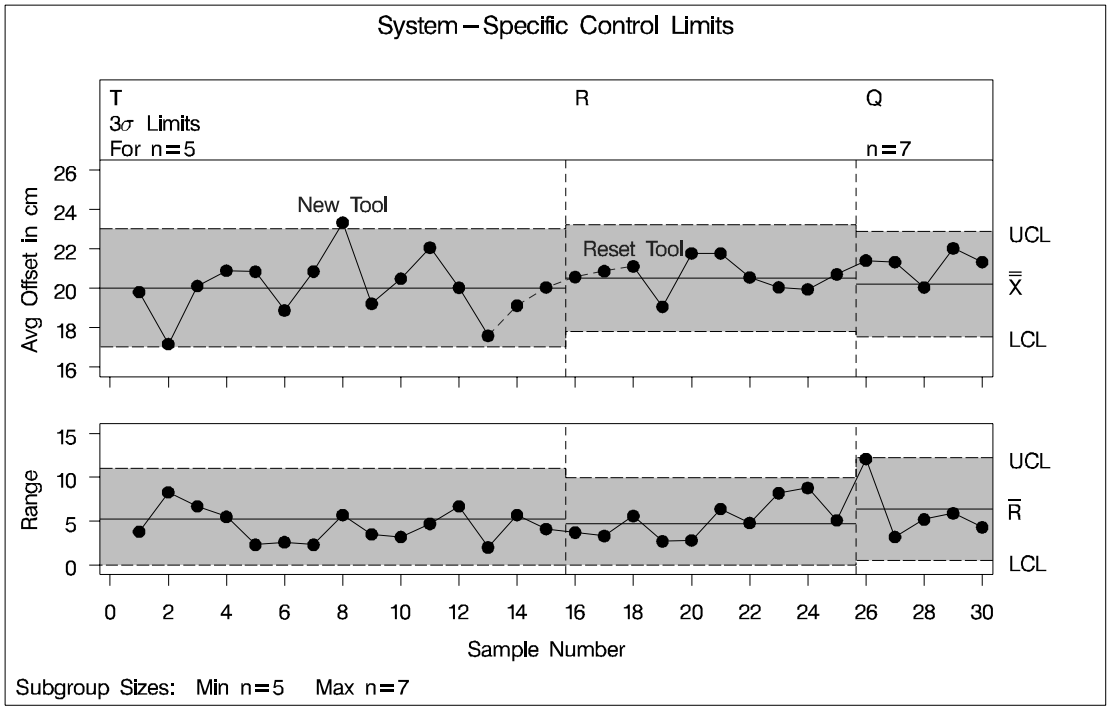


Figure 55.9. Multiple Sets of Control Limits with the TESTACROSS Option

Here, it is necessary to specify TESTNMETHOD=STANDARDIZE in conjunction with the TESTACROSS option, since the subgroup sample sizes are not constant across phases.

Although Test 3 is now signaled at sample 18, this result should be interpreted with care since the test is applied to standardized average offsets, and the averages for samples 13, 14, and 15 are standardized differently than the averages for samples 16, 17, and 18. If, for instance, the value of _MEAN_ for phase `R` in SYSLIM were 21.0 rather than 20.5, the standardized mean for sample 16 would be less than the standardized mean for sample 15, and Test 3 would not be signaled at sample 18.

In summary, when working with multiple control limits, you should

- use the TESTACROSS option only if the process is operating in a continuous manner across phases
- use TESTNMETHOD=STANDARDIZE only if it is clearly understood by users that tests signaled on the chart are based on *standardized* statistics rather than the plotted statistics

Enhancing the Display of Signaled Tests

There are various options for labeling points at which a test is signaled.

- The default label for Test *i* is *Testi*. See Figure 55.3 on page 1981 for an example.
- Specify TESTLABEL=SPACE to request labels of the form *Test i*. See Figure 55.4 on page 1984 for an example.
- Specify TESTLABEL*i='label'* to provide a specific *label* for the i^{th} test. See Figure 55.13 on page 1997 for an example.
- Specify TESTLABEL=(*variable*) to request labels provided by a *variable* in the input data set. See Figure 55.6 on page 1986 for an example.

If two or more tests are signaled at a particular point, the label displayed corresponds to the test that was specified first in the TESTS= list.

If you are using a graphics device, you can specify the color of the label and the connecting line segments for the pattern with the CTESTS= option. You can specify the line type for the line segments with the LTESTS= option; see Figure 55.3 on page 1981. If you are using a line printer, you can specify the plot character for the line segments with the TESTCHAR= option.

You can specify the ZONES option to display the zone lines on the chart, and you can specify ZONELABELS to label the zone lines. If you are using a graphics device, you can specify the color of the lines with the CZONES= option, and if you are using a line printer, you can specify the plot character for the lines with the ZONECHAR= option.

Nonstandard Tests for Special Causes

This section describes options and programming techniques for requesting various nonstandard tests for special causes.

Applying Tests to Range and Standard Deviation Charts

See SHWTSC3 in the SAS/QC Sample Library

If you are using the MRCHART, RCHART, SCHART, XRCHART, or XSCHART statement, you can use the TESTS2= option to request tests for special causes with an R chart or s chart. The syntax and test definitions for the TESTS2= option are identical to those for the TESTS= option, and you can use the ZONES2 and ZONE2LABELS options to display the zones on the secondary chart.

The following statements request Test 1 for a range chart of the data in ASSEMBLY (see page 1979):

```
title 'Analysis of Offset Ranges';
proc shewhart history=assembly;
   rchart offset * sample / sigma0    = 2.24
                            limitn    = 5
                            alln
                            tests2    = 1
                            testlabel = (comment) ;
   label offsetr = 'Offset Range in cm';
run;
```

The R chart is shown in Figure 55.10.

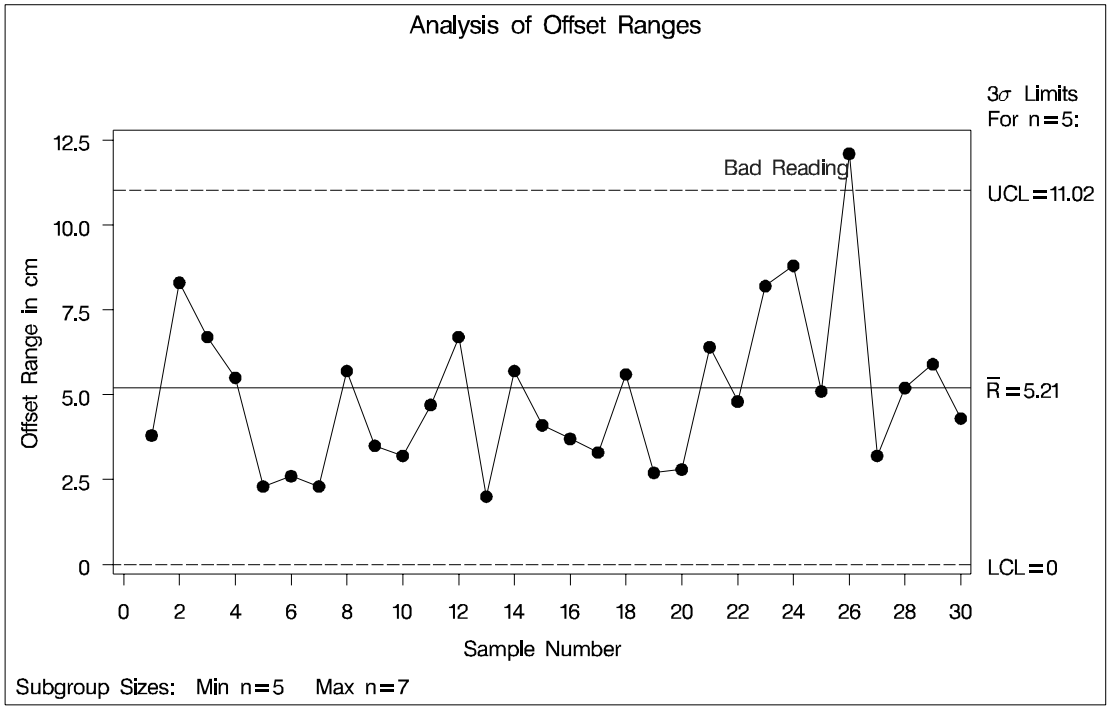


Figure 55.10. Range Chart with Test 1

CAUTION: Except for requesting Test 1, use of the TESTS2= option is not recommended for general process control work. At the time of this writing, there is insufficient published research supporting the application of the other tests to R charts and s charts. There are no established guidelines for interpreting the other tests, nor are there assessments of their false signal probabilities or average run length characteristics. The TESTS2= option is intended primarily as a research tool.

Applying Tests Based on Generalized Patterns

In addition to *indices* for standard tests, you can specify up to eight *T-patterns* or *M-patterns* with the TESTS= option:

- Specifying a T-pattern requests a search for k out of m points in a row in the interval (a, b). Tests based on T-patterns are generalizations of Tests 1, 2, 5, and 6. The average run length properties of tests based on T-patterns have been analyzed by Champ and Woodall (1987). Also refer to Chapter 8 of Wetherill and Brown (1991).
- Specifying an M-pattern requests a search for k points in a row increasing or decreasing. Tests based on M-patterns are generalizations of Test 3.

The general syntax for a T-pattern is of the form

T(K=*k* **M=***m* **LOWER=***a* **UPPER=***b* **SCHEME=***scheme* **CODE=***character* **LABEL=***'label'* **LEGEND=***'legend'* **)**

The options for a T-pattern are summarized in the following table:

Table 55.3. Options for T-Patterns

Option	Description
K=*k*	number of points $(k \leq m)$
M=*m*	number of consecutive points
LOWER=*value*	lower limit of interval (a, b)
UPPER=*value*	upper limit of interval (a, b)
SCHEME=ONESIDED	one-sided scheme using (a, b)
SCHEME=TWOSIDED	two-sided scheme using $(a, b) \cup (-b, -a)$
CODE=*character*	identifier for test (A-H)
LABEL=*'label'*	label for points that are signaled
LEGEND=*'legend'*	legend used with the TABLELEGEND option

The following rules apply to the T-pattern options:

1. You must specify SCHEME=*scheme*. Specifying SCHEME=ONESIDED requests a one-sided test that searches for k out of m points in a row in the interval (a, b). Specifying SCHEME=TWOSIDED with positive values for a and b (where $a < b$) requests a two-sided test that searches for k out of m points in a row in the interval (a, b) or k out of m points in a row in the interval $(-b, -a)$.

2. The values a and b must be specified in standardized units, and they must both have the same sign. For instance, specifying LOWER=2 and UPPER=3 with SCHEME=TWOSIDED corresponds to Zone A in Figure 55.1 on page 1978.
3. Specifying a missing value for the LOWER= option and a negative value for b requests a search in the interval $(-\infty, b)$. Specifying a positive value for a and a missing value for the UPPER= option requests a search in the interval (a, ∞).
4. You must specify a CODE= *character*, which can be any of the letters A through H. The character identifies the pattern in tables requested with the TABLETESTS and TABLEALL options and in the value of the variable _TESTS_ in the OUTTABLE= data set. The character is analogous to the indices 1 through 8 that are used to identify the standard tests. If you request multiple T-patterns, you must specify a unique character for each pattern.
5. You can specify a *label* with the LABEL= option. The label must be enclosed in quotes and can be up to 16 characters long. The label is used to label points on the chart at which the test defined by the T-pattern is signaled. The LABEL= option is similar to the TESTLABELn= options used with the standard tests.
6. You must specify a *legend* with the LEGEND= option if you also specify the TABLELEGEND or TABLEALL option. The legend must be enclosed in quotes and can be up to 40 characters long. The legend is used to describe the test defined by the T-pattern in the table legend requested with the TABLELEGEND and TABLEALL options.

An example of a nonstandard test using a T-pattern is the run test based on 14 out of 17 points in a row on the same side of the central line that is suggested by Wheeler and Chambers (1986). The following statements apply this test with Tests 1, 3, and 4. The resulting chart is shown in Figure 55.11.

See SHWTSC4 in the SAS/QC Sample Library

```
title 'Analysis of Assembly Data';
proc shewhart history=assembly;
   xrchart offset * sample /
      mu0       = 20
      sigma0    = 2.24
      limitn    = 5
      alln
      tests     = 1
                  t( k=14 m=17
                     lower=0 upper=. scheme=twosided
                     code=A label='Test A' )
                  3 4
      vaxis     = 16 to 26 by 2
      split     = '/' ;
   label offsetx = 'Avg Offset in cm/Range';
run;
```

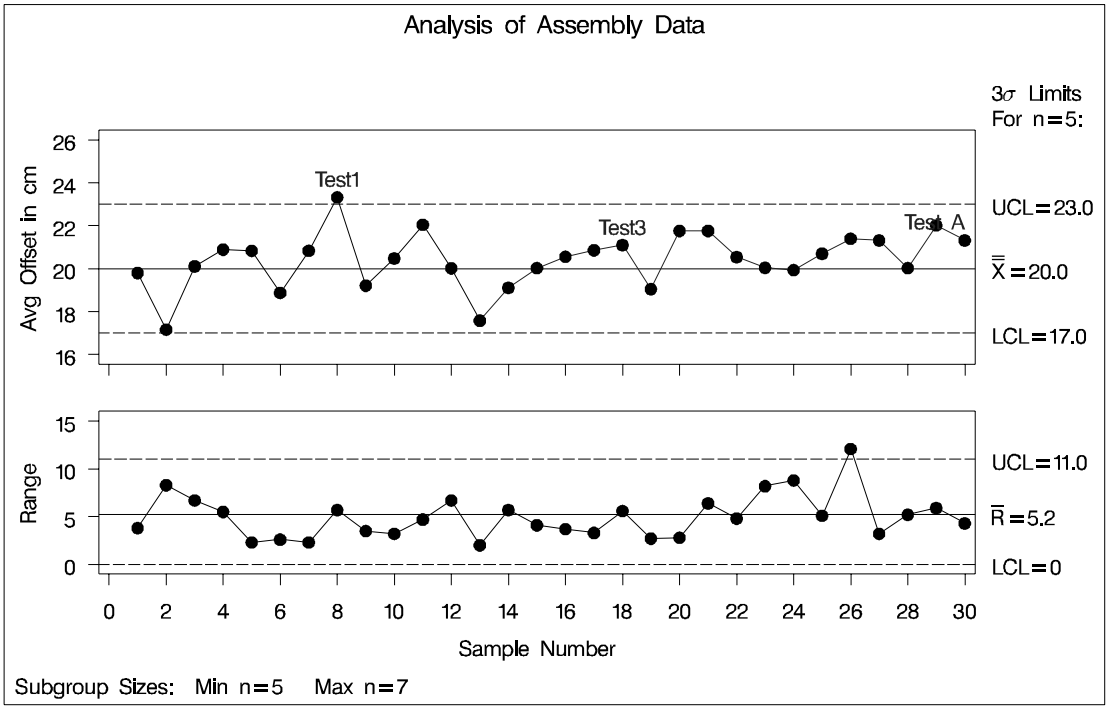


Figure 55.11. Generalized T-pattern Applied to Assembly Data

The specified T-pattern is signaled at 30^{h} subgroup. Consequently, this point is labeled *Test A*.

The general syntax for an M-pattern is of the form

M(K=*k* **DIR=***direction* **CODE=***character* **LABEL=***'label'* **LEGEND=***'legend'* **)**

The options for an M-pattern are summarized in the following table:

Table 55.4. Options for M-Patterns

Option	Description
K=*k*	number of points
DIR=INC	increasing pattern
DIR=DEC	decreasing pattern
CODE=*character*	identifier for test (A-H)
LABEL=*'label'*	label for points that are signaled
LEGEND=*'legend'*	legend used with the TABLELEGEND option

You must specify the direction of the pattern with the DIR= option. The CODE=, LABEL=, and LEGEND= options are used as described on page 1993.

CAUTION: You should not substitute tests based on arbitrarily defined T-patterns and M-patterns for standard tests in general process control applications. The pattern options are intended primarily as a research tool.

See SHWARL2 in the SAS/QC Sample Library

Champ and Woodall (1990) provide a FORTRAN program for assessing the run

length distribution of tests based on T-patterns. A version of their algorithm is implemented by a SAS/IML program in the SAS/QC Sample Library.

If you specify either a T-pattern or M-pattern with the TESTS= option and save the results in an OUTTABLE= data set, the length of the variable _TESTS_ is 16 rather than 8 (the default). The ninth character of _TESTS_ is assigned the value `A` if the test with CODE=A is signaled, the tenth character of _TESTS_ is assigned the value `B` if the test with CODE=B is signaled, and so on. If you also specify one or more standard tests, the t^{th} character of _TESTS_ is assigned the value i if Test i is signaled.

Customizing Tests with DATA Step Programs

Occasionally, you may find it necessary to apply customized tests that cannot be specified with the TESTS= option. You can program your own tests as follows:

See SHWTSC5 in the SAS/QC Sample Library

1. Run the SHEWHART procedure without the TESTS= option and save the results in an OUTTABLE= data set. Use the NOCHART option to suppress the display of the chart.
2. Use a DATA step program to apply your tests to the subgroup statistics in the OUTTABLE= data set. If tests are signaled at certain subgroups, save these results as values of a flag variable named _TESTS_, which should be a character variable of length 8. Recall that each observation of an OUTTABLE= data set corresponds to a subgroup. Assign the character *i* to the t^{th} character of _TESTS_ if the t^{th} customized test is signaled at that subgroup (otherwise, assign a blank character).
3. Run the procedure reading the modified data set as a TABLE= data set.

The following example illustrates these steps by creating an $\bar{X}$ chart for the data in ASSEMBLY (see "Requesting Standard Tests" on page 1979) that signals a special cause of variation if an average is greater than 2.5 standard errors above the central line. The first step is to compute 2.5σ limits and save both the subgroup statistics and the limits in an OUTTABLE= data set named FIRST.

```
proc shewhart history=assembly;
   xchart offset * sample /
      sigmas   = 2.5
      outtable = first
      nochart ;
run;


title ;
proc print data=first noobs;
run;
```

A partial listing of the data set FIRST is shown in Figure 55.12.

VAR	sample	_SIGMAS_	_LIMITN_	_SUBN_	_LCLX_	_SUBX_	_MEAN_	_UCLX_	_EXLIM_
offset	1	2.5	5	5	18.1515	19.80	20.4733	22.7951	
offset	2	2.5	5	5	18.1515	17.16	20.4733	22.7951	LOWER
offset	3	2.5	5	5	18.1515	20.11	20.4733	22.7951	
offset	4	2.5	5	5	18.1515	20.89	20.4733	22.7951	
offset	5	2.5	5	5	18.1515	20.83	20.4733	22.7951	
offset	6	2.5	5	5	18.1515	18.87	20.4733	22.7951	
offset	7	2.5	5	5	18.1515	20.84	20.4733	22.7951	
offset	8	2.5	5	5	18.1515	23.33	20.4733	22.7951	UPPER
offset	9	2.5	5	5	18.1515	19.21	20.4733	22.7951	
offset	10	2.5	5	5	18.1515	20.48	20.4733	22.7951	
.	.	.	.	.	.	.	.	.	
.	.	.	.	.	.	.	.	.	
.	.	.	.	.	.	.	.	.	
offset	30	2.5	7	7	18.5111	21.32	20.4733	22.4356	

Figure 55.12. Partial Listing of the Data Set FIRST

The second step is to carry out the test and create the flag variable _TESTS_.

```
data first;
   set first;
   length _tests_ $ 8;
   if _subx_ > _uclx_ then substr( _tests_, 1 ) = '1';
run;
```

Finally, the data set FIRST is read by the SHEWHART procedure as a TABLE= data set.

```
title 'Customized Analysis of Assembly Data';
proc shewhart table=first;
   xchart offset * sample / tests      = 1
                            testlabel1 = 'Test Signaled';
   label _subx_ = 'Average Offset in cm';
run;
```

The chart is shown in Figure 55.13. Note that the variable _TESTS_ is read "as is" to flag points on the chart, and the standard tests are *not* applied to the data. The option TESTS=1 specifies that a point is to be labeled if the first character of _TESTS_ for the corresponding subgroup is `1`. The label is specified by the TESTLABEL1= option (the default would be *Test1*).

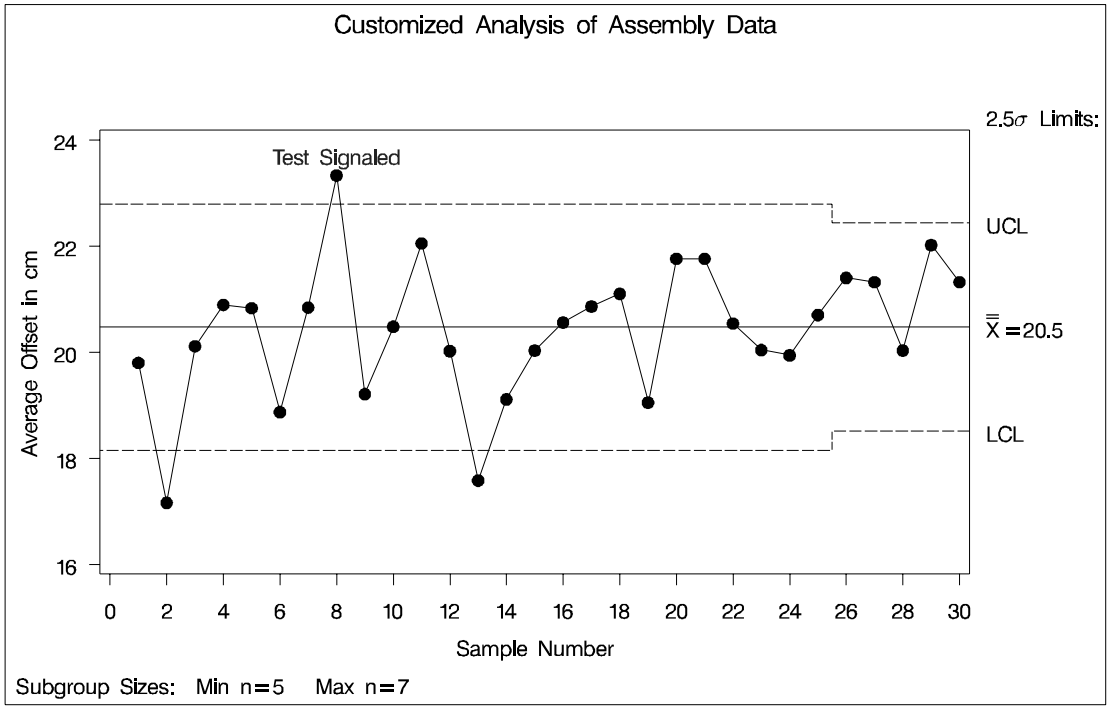


Figure 55.13. Customized Test

In general, you can simultaneously apply up to eight customized tests with the variable _TESTS_, which is of length 8. If two or more tests are signaled at a particular point, the label that is displayed corresponds to the test that appears first in the TESTS= list. In the preceding example, the test involves only the current subgroup. For customized tests involving patterns that span multiple subgroups, you will find it helpful to use the LAG functions described in *SAS Language Reference: Dictionary*.

Notes:

1. If you provide the variable _TESTS_ in a TABLE= data set, you must also use the TESTS= option to specify which characters of _TESTS_ are to be checked.
2. The CTESTS= and LTESTS= options specify colors and line styles for *standard* patterns and may not be applicable with customized tests.

Chapter 56
Specialized Control Charts

Chapter Contents

OVERVIEW 2001

AUTOCORRELATION IN PROCESS DATA 2001
Diagnosing and Modeling Autocorrelation 2003
Strategies for Handling Autocorrelation 2005

MULTIPLE COMPONENTS OF VARIATION 2009
Preliminary Examination of Variation 2010
Determining the Components of Variation 2013

SHORT RUN PROCESS CONTROL 2016
Analyzing the Difference from Nominal 2018
Testing for Constant Variances 2025
Standardizing Differences from Nominal 2026

NONNORMAL PROCESS DATA 2027
Creating a Preliminary Individual Measurements Chart 2028
Calculating Probability Limits 2030

MULTIVARIATE CONTROL CHARTS 2033
Calculating the Chart Statistic 2033
Examining the Principal Component Contributions 2036

Chapter 56
Specialized Control Charts

Overview

Although the Shewhart chart serves well as the fundamental tool for statistical process control (SPC) applications, its assumptions are challenged by many modern manufacturing environments. For example, when standard control limits are used in applications where the process is sampled frequently, autocorrelation in the measurements can result in too many out-of-control signals. This chapter also considers process control applications involving multiple components of variation, short production runs, nonnormal process data, and multivariate process data.

These questions are subjects of current research and debate. It is not the goal of this chapter to provide definitive solutions but rather to illustrate some basic approaches that have been proposed and indicate how they can be implemented with short SAS programs. The sections in this chapter use the SHEWHART procedure in conjunction with various SAS procedures for statistical modeling, as summarized by the following table:

Process Control Application	Modeling Procedure
Diagnosing and modeling autocorrelation in process data	ARIMA
Developing control limits for processes involving multiple components of variation	MIXED
Establishing control with short production runs and checking for constant variance	GLM
Developing control limits for nonnormal individual measurements	CAPABILITY
Creating control charts for multivariate process data	PRINCOMP

Autocorrelation in Process Data

See SHWARIEW in the SAS/QC Sample Library

Autocorrelation has long been recognized as a natural phenomenon in process industries, where parameters such as temperature and pressure vary slowly relative to the rate at which they are measured. Only in recent years has autocorrelation become an issue in SPC applications, particularly in parts industries, where autocorrelation is viewed as a problem that can undermine the interpretation of Shewhart charts. One reason for this concern is that, as automated data collection becomes prevalent in parts industries, processes are sampled more frequently and it is possible to recognize autocorrelation that was previously undetected. Another reason, noted by Box and Kramer (1992), is that the distinction between parts and process industries is becom-

ing blurred in areas such as computer chip manufacturing. For two other discussions of this issue, refer to Schneider and Pruett (1994) and Woodall (1993).

The standard Shewhart analysis of individual measurements assumes that the process operates with a constant mean μ, and that x_t (the measurement at time t) can be represented as $x_t = \mu + \epsilon_t$, where ϵ_t is a random displacement or error from the process mean μ. Typically, the errors are assumed to be statistically independent in the derivation of the control limits displayed at three standard deviations above and below the central line, which represents an estimate for μ.

When Shewhart charts are constructed from autocorrelated measurements, the result can be too many false signals, making the control limits seem too tight. This situation is illustrated in Figure 56.1, which displays an individual measurement and moving range chart for 100 observations of a chemical process.

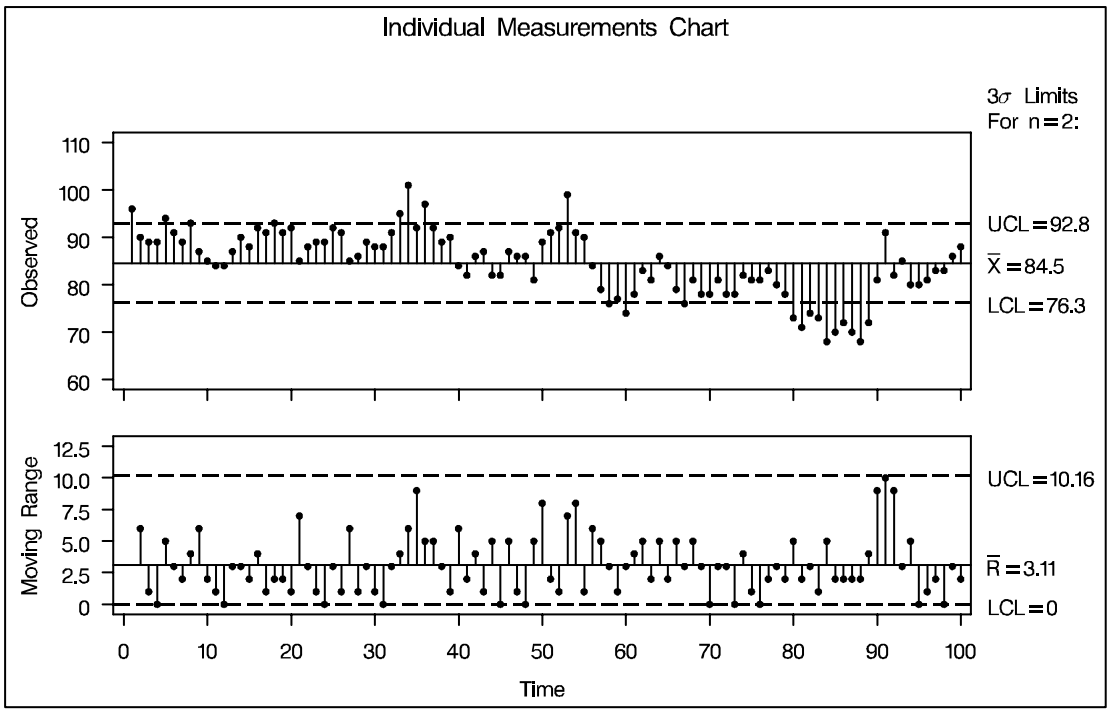


Figure 56.1. Conventional Shewhart Chart

The measurements are saved in a SAS data set named CHEMICAL.* The chart in Figure 56.1 is created with the following statements:

*The measurements are patterned after the values plotted in Figure 1 of Montgomery and Mastrangelo (1991).

```
goptions lfactor=3;
symbol h=2.0 pct;
title 'Individual Measurements Chart';
proc shewhart data=chemical;
   irchart xt*t / cneedles  = black
                  npanelpos = 100
                  split     = '/';
   label xt = 'Observed/Moving Range'
         t = 'Time';
run;
```

Diagnosing and Modeling Autocorrelation

You can diagnose autocorrelation with an autocorrelation plot created with the ARIMA procedure.

```
title ;
proc arima data=chemical;
   identify var = xt;
run;
```

Refer to *SAS/ETS User's Guide* for details on the ARIMA procedure. The plot, shown in Figure 56.2, indicates that the data are highly autocorrelated with a lag 1 autocorrelation of 0.83.

The partial autocorrelation plot in Figure 56.2 suggests that the data can be modeled with a first-order autoregressive model, commonly referred to as an AR(1) model.

$$\tilde{x}_t \equiv x_t - \mu = \phi_0 + \phi_1 \tilde{x}_{t-1} + \epsilon_t$$

You can fit this model with the ARIMA procedure. The results in Figure 56.3 show that the equation of the fitted model is $\tilde{x}_t = 13.05 + 0.847\tilde{x}_{t-1}$.

```
proc arima data=chemical;
   identify var=xt;
   estimate p=1 method=ml;
run;
```

The ARIMA Procedure

Autocorrelations

Lag	Covariance	Correlation	-1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1
0	48.348400	1.00000	|********************|
1	40.141884	0.83026	|*****************. |
2	34.732168	0.71837	|**************. |
3	29.950852	0.61948	|************. |
4	24.739536	0.51169	|**********. |
5	20.594420	0.42596	|*********. |
6	18.427704	0.38114	|********. |
7	17.400188	0.35989	|******* . |
8	17.621272	0.36446	|******* . |
9	18.363756	0.37982	|******** . |
10	16.754040	0.34653	|******* . |
11	16.844924	0.34841	|******* . |
12	17.137208	0.35445	|******* . |
13	16.884092	0.34922	|******* . |
14	17.927976	0.37081	|******* . |
15	16.801860	0.34752	|******* . |
16	17.076544	0.35320	|******* . |
17	17.815028	0.36847	|******* . |
18	16.501312	0.34130	|******* . |
19	14.662196	0.30326	|****** . |
20	12.612280	0.26086	|***** . |
21	11.105364	0.22969	|***** . |
22	8.891648	0.18391	|**** . |
23	6.794132	0.14052	|*** . |
24	4.732816	0.09789	|** . |

"." marks two standard errors

Partial Autocorrelations

Lag	Correlation	-1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1
1	0.83026	| . |***************** |
2	0.09346	| . |** . |
3	0.00385	| . | . |
4	-0.07340	| . *| . |
5	-0.00278	| . | . |
6	0.09013	| . |** . |
7	0.08781	| . |** . |
8	0.10327	| . |** . |
9	0.07240	| . |* . |
10	-0.11637	| . **| . |
11	0.08210	| . |** . |
12	0.07580	| . |** . |
13	0.04429	| . |* . |
14	0.11661	| . |** . |
15	-0.10446	| . **| . |
16	0.07703	| . |** . |
17	0.07376	| . |* . |
18	-0.07080	| . *| . |
19	-0.02814	| . *| . |
20	-0.08559	| . **| . |
21	0.01962	| . | . |
22	-0.04599	| . *| . |
23	-0.07878	| . **| . |
24	-0.02303	| . | . |

Figure 56.2. Autocorrelation Plots for Chemical Data

The ARIMA Procedure

Maximum Likelihood Estimation

Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	85.28375	2.32973	36.61	<.0001	0
AR1,1	0.84694	0.05221	16.22	<.0001	1

Constant Estimate	13.05329
Variance Estimate	14.27676
Std Error Estimate	3.77846
AIC	552.8942
SBC	558.1045
Number of Residuals	100

Figure 56.3. Fitted AR(1) Model

Strategies for Handling Autocorrelation

There is considerable disagreement on how to handle autocorrelation in process data. Consider the following three views:

- At one extreme, Wheeler (1991b) argues that the usual control limits are contaminated "only when the autocorrelation becomes excessive (say 0.80 or larger)." He concludes that "one need not be overly concerned about the effects of autocorrelation upon the control chart."
- At the opposite extreme, automatic process control (APC), also referred to as engineering process control, views autocorrelation as a phenomenon to be exploited. In contrast to SPC, which assumes that the process remains on target unless an unexpected but removable cause occurs, APC assumes that the process is changing dynamically due to known causes that cannot be eliminated. Instead of avoiding "overcontrol" and "tampering," which have a negative connotation in the SPC framework, APC advocates continuous tuning of the process to achieve minimum variance control. Descriptions of this approach and discussion of the differences between APC and SPC are provided by a number of authors, including Box and Kramer (1992), MacGregor (1987, 1990), MacGregor, Hunter, and Harris (1988), and Montgomery and others (1994).
- A third strategy advocates removing autocorrelation from the data and constructing a Shewhart chart (or an EWMA chart or a cusum chart) for the residuals; refer, for example, to Alwan and Roberts (1988).

An example of the last approach is presented in the remainder of this section simply to demonstrate the use of the ARIMA procedure in conjunction with the SHEWHART procedure. The ARIMA procedure models the autocorrelation and saves the residuals in an output data set; the SHEWHART procedure creates a control chart using the residuals as input data.

In the chemical data example, the residuals can be computed as forecast errors and saved in an output SAS data set with the FORECAST statement in the ARIMA procedure.

```
proc arima data=chemical;
   identify var=xt;
   estimate p=1 method=ml;
   forecast out=results id=t;
run;
```

The output data set (named RESULTS) saves the one-step-ahead forecasts as a variable named FORECAST, and it also contains the original variables XT and T. You can create a Shewhart chart for the residuals by using the data set RESULTS as input to the SHEWHART procedure.

```
title 'Residual Analysis Using AR(1) Model';
proc shewhart data=results(firstobs=4 obs=100);
   xchart xt*t / npanelpos = 100
                 split     = '//'
                 trendvar  = forecast
                 xsymbol   = xbar
                 ypct1     = 40
                 vref2     = 70 to 100 by 10
                 lvref     = 2
                 nolegend;
   label xt = 'Residual/Forecast'
         t = 'Time';
run;
```

The chart is shown in Figure 56.4. Specifying TRENDVAR=FORECAST plots the values of FORECAST in the lower chart and plots the residuals (XT − FORECAST) together with their 3σ limits in the upper chart.*

Various other methods can be applied with this data. For example, Montgomery and Mastrangelo (1991) suggest fitting an exponentially weighted moving average (EWMA) model and using this model as the basis for a display that they refer to as an *EWMA central line control chart*.

Before presenting the statements for creating this display, it is helpful to review some terminology. The EWMA *statistic* plotted on a conventional EWMA control chart is defined as

$$z_t = \lambda x_t + (1 - \lambda) z_{t-1}$$

*The upper chart in Figure 56.4 resembles Figure 2 of Montgomery and Mastrangelo (1991), who conclude that the process is in control.

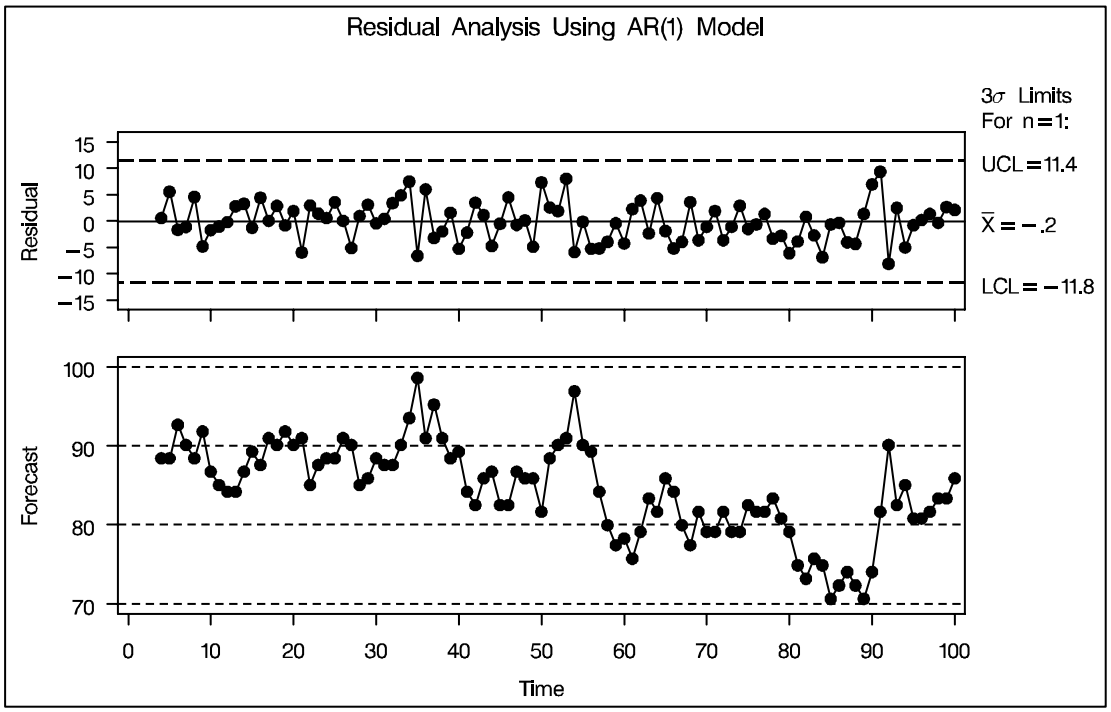


Figure 56.4. Residuals from AR(1) Model

The EWMA chart (which you can construct with the MACONTROL procedure) is based on the assumption that the observations x_t are independent. However, in the context of autocorrelated process data (and more generally in time series analysis), the EWMA statistic z_t plays a different role:* it is the optimal one-step-ahead forecast for a process that can be modeled by an ARIMA(0,1,1) model

$$x_t = x_{t-1} + \epsilon_t - \theta\epsilon_{t-1}$$

provided that the weight parameter λ is chosen as $\lambda = 1 - \theta$. This statistic is also a good predictor when the process can be described by a subset of ARIMA models for which the process is "positively autocorrelated and the process mean does not drift too quickly."†

You can fit an ARIMA(0,1,1) model to the chemical data with the following statements. A summary of the fitted model is shown in Figure 56.5.

```
title ;
proc arima data=chemical;
   identify var=xt(1);
   estimate q=1 method=ml noint;
   forecast out=ewma id=t;
run;
```

*For a discussion of these roles, refer to Hunter (1986).
†Refer to Montgomery and Mastrangelo (1991) and the discussion that follows their paper.

```
                        The ARIMA Procedure

                   Maximum Likelihood Estimation

                                 Standard                  Approx
Parameter       Estimate            Error    t Value    Pr > |t|     Lag

MA1,1            0.15041          0.10021       1.50      0.1334       1


                  Variance Estimate      14.97024
                  Std Error Estimate      3.86914
                  AIC                    549.868
                  SBC                   552.4631
                  Number of Residuals         99
```

Figure 56.5. Fitted ARIMA(0, 1, 1) Model

The forecast values and their standard errors (variables FORECAST and STD), together with the original measurements, are saved in a data set named EWMA. The EWMA central line control chart plots the forecasts from the ARIMA(0,1,1) model as the central "line," and it uses the standard errors of prediction to determine upper and lower control limits. You can construct this chart, shown in Figure 56.6,* with the following statements:

```
data ewma;
   set ewma(firstobs=2 obs=100);
run;

data ewmatab;
   length _var_ $ 8 ;
   set ewma (rename=(forecast=_mean_ xt=_subx_));
   _var_    = 'xt';
   _sigmas_ = 3;
   _limitn_ = 1;
   _lclx_   = _mean_ - 3 * std;
   _uclx_   = _mean_ + 3 * std;
   _subn_   = 1;
run;


title 'EWMA Center Line Control Chart';
proc shewhart table=ewmatab;
   xchart xt*t / npanelpos = 100
                 xsymbol   = 'Center'
                 cinfill   = ligr
                 llimits   = 1
                 nolegend;
   label _subx_ = 'Observed'
               t = 'Time' ;
run;
```

*Figure 56.6 is similar to Figure 5 of Montgomery and Mastrangelo (1991).

Note that EWMA is read by the SHEWHART procedure as a TABLE= input data set, which has a special structure intended for applications in which both the statistics to be plotted and their control limits are pre-computed. The variables in a TABLE= data set have reserved names beginning and ending with the underscore character; for this reason, FORECAST and XT are temporarily renamed as _MEAN_ and _SUBX_, respectively. For more information on TABLE= data sets, see "Input Data Sets" in the chapter for the chart statement in which you are interested.

Again, the conclusion is that the process is in control. While Figure 56.4 and Figure 56.6 are not the only displays that can be considered for analyzing the chemical data, their construction illustrates the conjunctive use of the ARIMA and SHEWHART procedures in process control applications involving autocorrelated data.

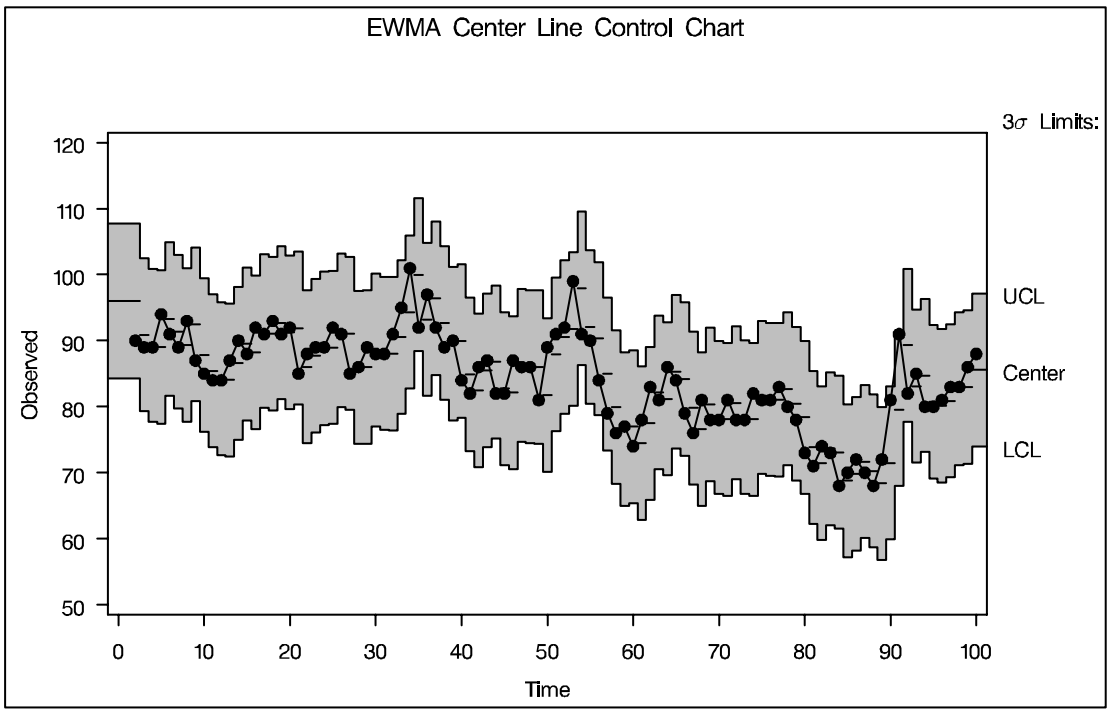


Figure 56.6. EWMA Center Line Chart

Multiple Components of Variation

See SHWMULTC in the SAS/QC Sample Library

In the preceding section, the excessive variation in the conventional Shewhart chart in Figure 56.1 is the result of positive autocorrelation in the data. The variation is "excessive" not because it is due to special causes of variation, but because the Shewhart model is inappropriate. This section considers another form of departure from the Shewhart model; here, measurements are *independent* from one subgroup sample to the next, but there are multiple components of variation for each measurement. This is illustrated with an example involving two components.*

*Also refer to Chapter 5 of Wheeler and Chambers (1986) for an explanation of the effects of subgrouping and sources of variation on control charts.

A company that manufactures polyethylene film monitors the statistical control of an extrusion process that produces a continuous sheet of film. At periodic intervals of time, samples are taken at four locations (referred to as lanes) along a cross section of the sheet, and a test measurement is made of each sample. The test values are saved in a SAS data set named FILM. A partial listing of FILM is shown in Figure 56.7.

sample	lane	testval
1	A	93
1	B	87
1	C	92
1	D	78
2	A	87

Figure 56.7. Polyethylene Sheet Measurements in the Data Set FILM

Preliminary Examination of Variation

As a preliminary step in the analysis, the data are sorted by lane and visually screened for outliers (test values greater than 130) with box plots created as follows:

```
proc sort data=film;
   by lane;
run;

title 'Outlier Analysis';
proc shewhart data=film;
   boxchart testval*lane / boxstyle = schematicid
                           idsymbol = dot
                           cboxfill = megr
                           vref     = 130
                           vreflab  = 'Outlier Cutoff'
                           hoffset  = 5
                           nolegend
                           stddevs
                           nolimits ;
   id sample;
run;
```

Specifying BOXSTYLE=SCHEMATICID requests schematic box plots with outliers identified by the value of the ID variable SAMPLE. The STDDEVS option specifies that the estimate of the process standard deviation is to be based on subgroup standard deviations. Although this estimate is not needed here because control limits are not displayed, it is recommended that you specify the STDDEVS option whenever you are working with subgroup sample sizes greater than ten. The NOLEGEND and NOLIMITS options suppress the subgroup sample size legend and control limits for lane means that are displayed by default. The display is shown in Figure 56.8.

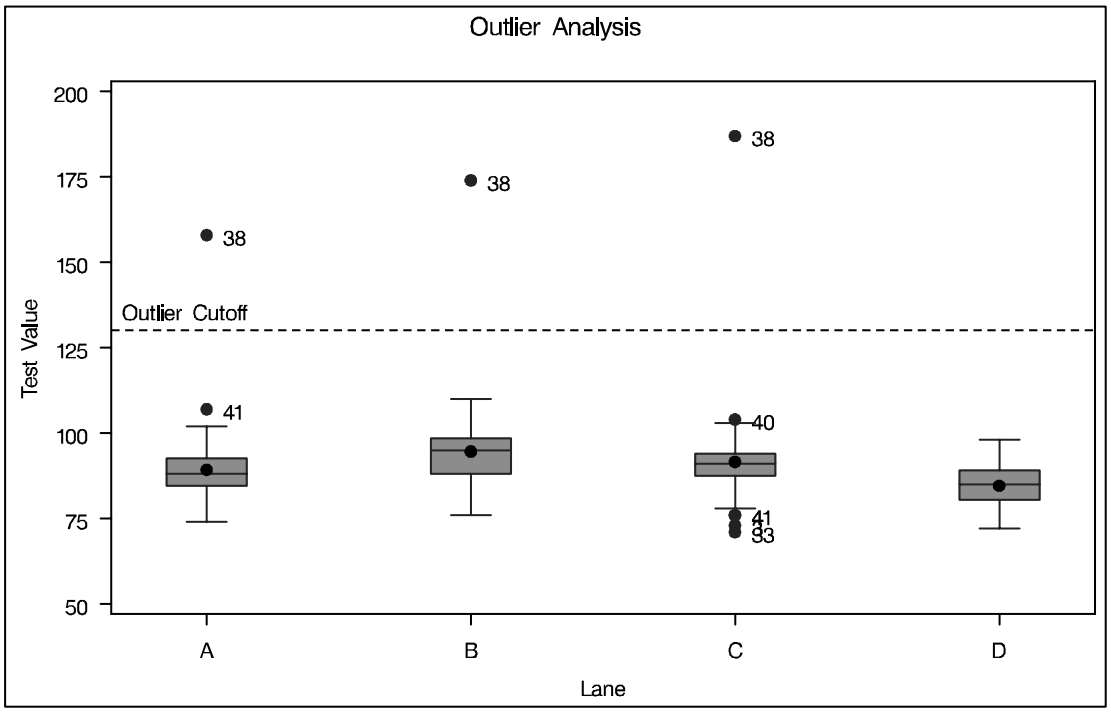


Figure 56.8. Outlier Analysis for the Data Set FILM

Figure 56.9 shows similarly created box plots for the data in FILM2, which is created by removing the outliers from the data set FILM.

```
data film2;
   set film;
   if testval < 130;

title 'Variation Within Lane';
proc shewhart data=film2;
   boxchart testval*lane / boxstyle = schematicid
                           boxwidth = 5
                           idsymbol = dot
                           cboxfill = megr
                           hoffset  = 5
                           nolegend
                           stddevs
                           nolimits ;
   id sample;
run;
```

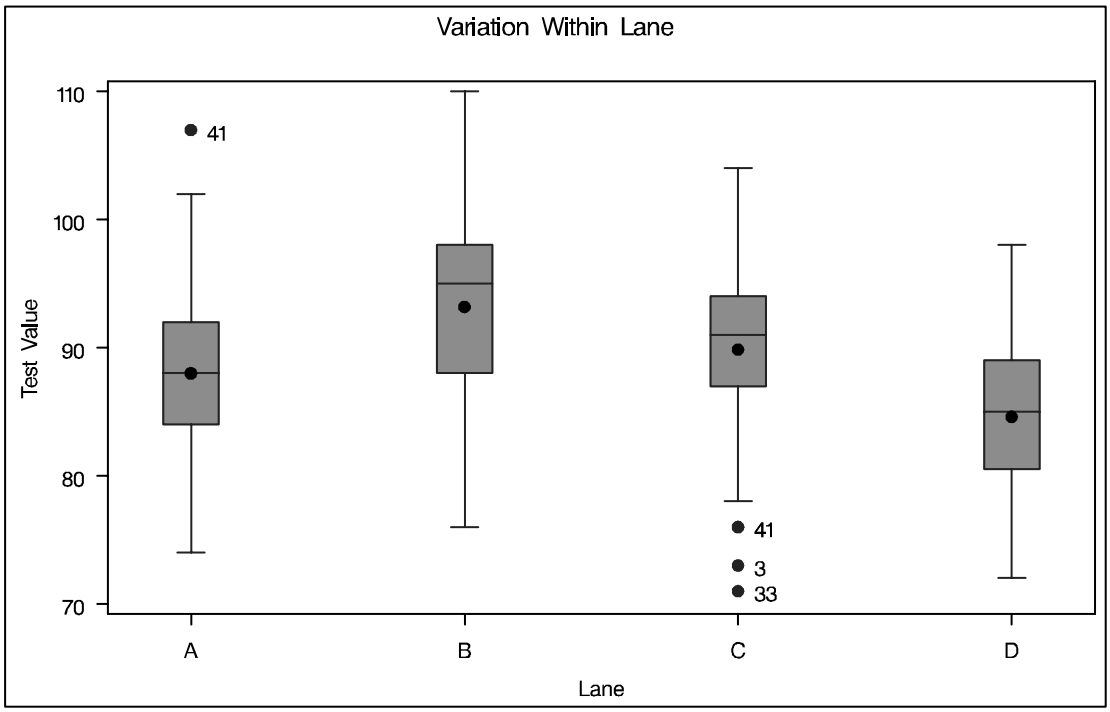


Figure 56.9. The Data Set FILM2 Without Outliers

Since you have no additional information about the process, you may want to create a conventional $\bar{X}$ and R chart for the test values grouped by the variable SAMPLE. This is a straightforward application of the XRCHART statement in the SHEWHART procedure.

```
proc sort data=film2;
   by sample;
run;

symbol h=2.0 pct;
title 'Shewhart Chart for Means and Ranges';
proc shewhart data=film2;
   xrchart testval*sample /
      split     = '/'
      npanelpos = 60
      limitn    = 4
      coutfill  = megr
      outlimits = rlimits
      nolegend
      alln;
   label testval='Average Test Value/Range';
run;
```

The $\bar{X}$ and R chart is displayed in Figure 56.10. Ordinarily, the out-of-control points in the $\bar{X}$ chart would indicate that the process is not in statistical control. In this

situation, however, the process is known to be quite stable, and the data have been screened for outliers. The problem is that the control limits for the average test value were computed from an inappropriate model. This is discussed in the following section.

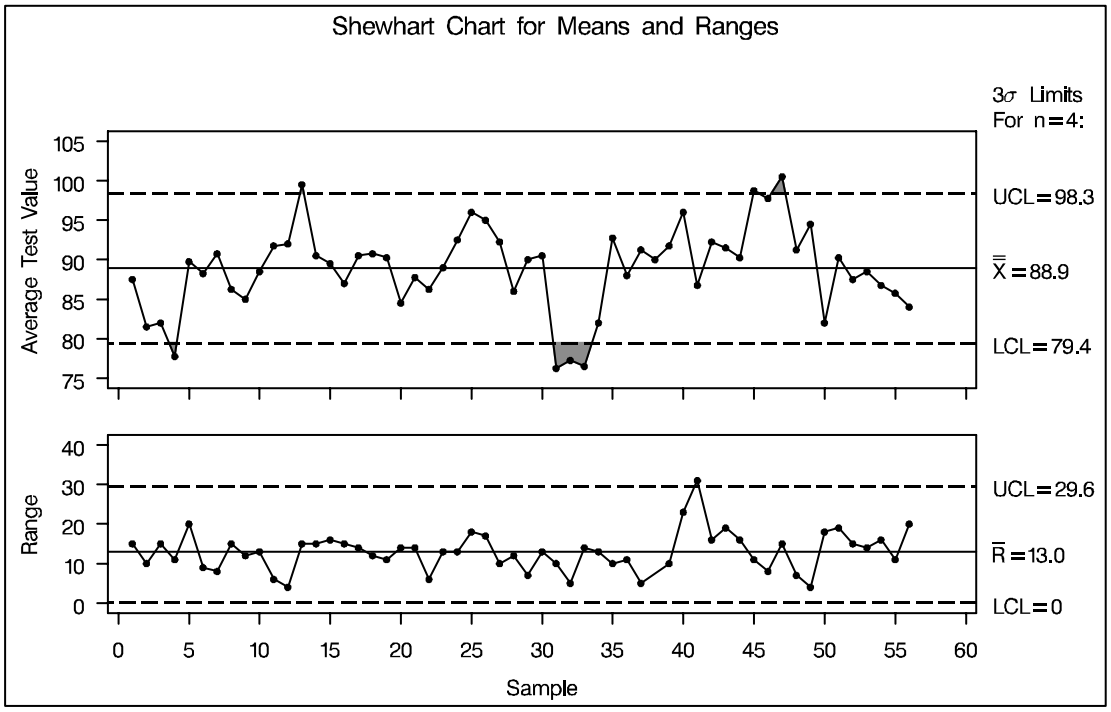


Figure 56.10. Conventional $\bar{X}$ and R Chart

Determining the Components of Variation

The standard Shewhart analysis assumes that sampling variation, also referred to as *within-group* variation, is the only source of variation. Writing x_{ij} for the jth measurement within the ith subgroup, you can express the model for the conventional $\bar{X}$ and R chart as

$$x_{ij} = \mu + \sigma_W \epsilon_{ij} \tag{1}$$

for $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, n$. The random variables ϵ_{ij} are assumed to be independent with zero mean and unit variance, and σ_W^2 is the within-subgroup variance. The parameter μ denotes the process mean.

In a process such as film manufacturing, this model is not adequate because there is additional variation due to changes in temperature, pressure, raw material, and other factors. A more appropriate model is

$$x_{ij} = \mu + \sigma_B \omega_i + \sigma_W \epsilon_{ij} \tag{2}$$

where σ_B^2 is the *between-subgroup* variance, the random variables ω_i are independent with zero mean and unit variance, and the random variables ω_i are independent of the random variables ϵ_{ij}.*

To plot the subgroup averages $\bar{x}_{i.} \equiv \frac{1}{n}\sum_{j=1}^{n} x_{ij}$ on a control chart, you need expressions for the expectation and variance of $\bar{x}_{i.}$. These are

$$\begin{aligned} E(\bar{x}_{i.}) &= \mu \\ \text{Var}(\bar{x}_{i.}) &= \sigma_B^2 + \frac{\sigma_W^2}{n} \end{aligned}$$

Thus, the central line should be located at $\hat{\mu}$, and 3σ limits should be located at

$$\hat{\mu} \pm 3\sqrt{\widehat{\sigma_B^2} + \frac{\widehat{\sigma_W^2}}{n}} \tag{3}$$

where $\widehat{\sigma_B^2}$ and $\widehat{\sigma_W^2}$ denote estimates of the variance components. You can use a variety of SAS procedures for fitting linear models to estimate the variance components. The following statements show how this can be done with the MIXED procedure:

```
title;
proc mixed data=film2;
   class sample;
   model testval = / s;
   random sample;
   ods output solutionf=sf;
   ods output covparms=cp;
run;
```

The results are shown in Figure 56.11. Note that the parameter estimates are $\widehat{\sigma_B^2} = 19.25$, $\widehat{\sigma_W^2} = 39.68$, and $\hat{\mu} = 88.90$.

The Mixed Procedure

Covariance Parameter Estimates

Cov Parm	Estimate
sample	19.2526
Residual	39.6825

Solution for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	88.8963	0.7250	55	122.61	<.0001

Figure 56.11. Partial Output from the MIXED Procedure

*This notation is used in Chapter 3 of Wetherill and Brown (1991), which discusses this issue.

The following statements merge the output data sets from the MIXED procedure into a SAS data set named NEWLIM that contains the appropriately derived control limit parameters for average test value:

```
data cp;
   set cp sf;
   keep Estimate;
run;

proc transpose data=cp out=newlim;
run;

data newlim (keep=_lclx_ _mean_ _uclx_);
  set newlim;
  _limitn_ = 4;
  _mean_   = col3;
  _stddev_ = sqrt(4*col1 + col2);
  _lclx_   = _mean_ - 3*_stddev_ / sqrt(_limitn_);
  _uclx_   = _mean_ + 3*_stddev_ / sqrt(_limitn_);
  output;
run;
```

Here, the variable _LIMITN_ is assigned the value of n, the variable _MEAN_ is assigned the value of $\hat{\mu}$, and the variable _STDDEV_ is assigned the value of

$$\hat{\sigma}_{\text{adj}} \equiv \sqrt{4\widehat{\sigma_B^2} + \widehat{\sigma_W^2}}$$

The 3σ limits (_LCLX_ and _UCLX_) are computed according to (3) using $\hat{\sigma}_{\text{adj}}$. The data set NEWLIM contains the mean and 3σ limits for the average test value.

The following statements compute appropriate control limits for the $\bar{X}$ and R charts, which are shown in Figure 56.12. First, the data set NEWLIM2 is created by merging the data set RLIMITS, which contains the original R chart limits computed in "Preliminary Examination of Variation", with NEWLIM, which saved the appropriate $\bar{X}$ chart limits. The original R chart limits are valid because the range in the ith subgroup is $R_i = \sigma_W(\max_j \epsilon_{ij} - \min_j \epsilon_{ij})$, which is the same for models (1) and (2). The LIMITS= option specifies the data set NEWLIM2 as the source of the control limits for Figure 56.12.

```
data newlim2;
   merge newlim rlimits (drop=_lclx_ _mean_ _uclx_);
run;

title 'Control Chart with Adjusted Limits';
proc shewhart data=film2 limits=newlim2;
   xrchart testval*sample / npanelpos = 60;
   label testval='Average Test Value';
run;
```

The control limits for the $\bar{X}$ chart in Figure 56.12 are $\hat{\mu} \pm \frac{3}{\sqrt{n}}\hat{\sigma}_{\text{adj}}$. This chart correctly indicates that the variation in the process is due to common causes.

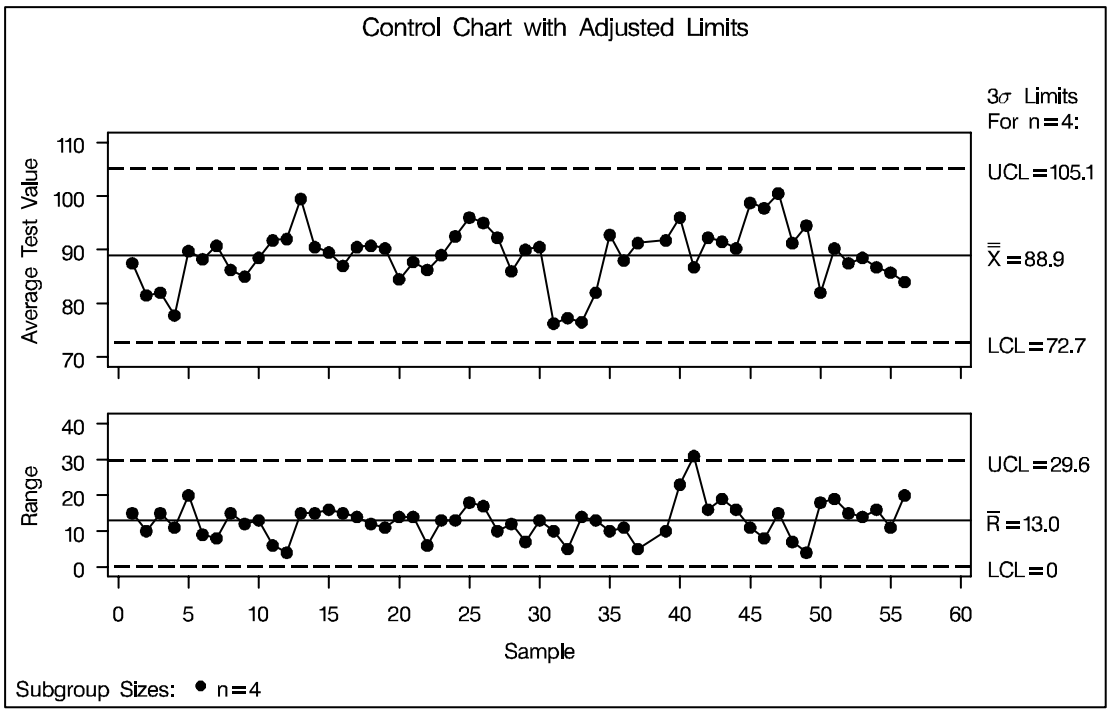


Figure 56.12. $\bar{X}$ and R Chart with Derived Control Limits

You can use a similar set of statements to display the derived control limits in NEWLIM on an $\bar{X}$ and R chart for the original data (including outliers), as shown in Figure 56.13.

A simple alternative to the chart in Figure 56.12 is an "individual measurements" chart for the subgroup means. The advantage of the variance components approach is that it yields separate estimates of the components due to lane and sample, as well as a number of hypothesis tests (these require assumptions of normality). In applying this method, however, you should be careful to use data that represent the process in a state of statistical control.

Short Run Process Control

See SHWSRUN in the SAS/QC Sample Library

When conventional Shewhart charts are used to establish statistical control, the initial control limits are typically based on 25 to 30 subgroup samples. Often, however, this amount of data is not available in manufacturing situations where product changeover occurs frequently or production runs are limited.

A variety of methods have been introduced for analyzing data from a process that is alternating between short runs of multiple products. The methods commonly used in the United States are variations of two basic approaches:*

*For a review of related methods, refer to Al-Salti and Statham (1994).

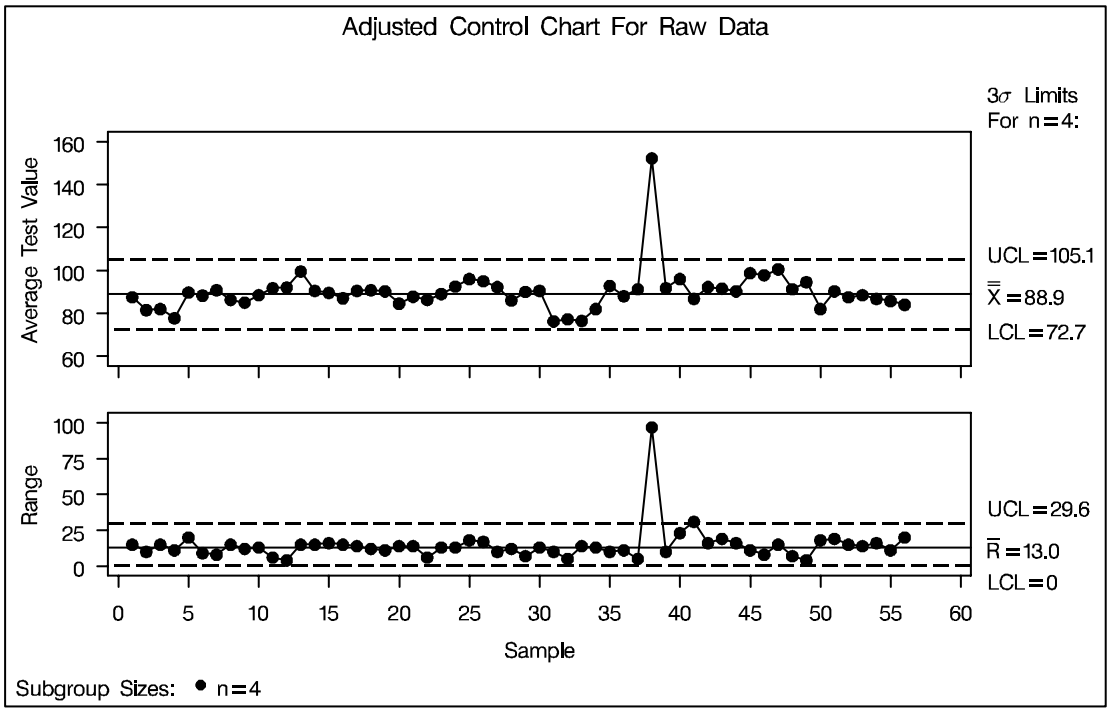


Figure 56.13. $\bar{X}$ and R Chart with Derived Control Limits for Raw Data

- the *difference from nominal* approach. A product-specific nominal value is subtracted from each measured value, and the differences (together with appropriate control limits) are charted. Here it is assumed that the nominal value represents the central location of the process (ideally estimated with historical data) and that the process variability is constant across products.
- the *standardization* approach. Each measured value is standardized with a product-specific nominal and standard deviation values. This approach is followed when the process variability is not constant across products.

These approaches are highlighted in this section because of their popularity, but two alternatives that are technically more sophisticated are worth noting.

- Hillier (1969) provided a method for modifying the usual control limits for $\bar{X}$ and R charts in startup situations where fewer than 25 subgroup samples are available for estimating the process mean μ and standard deviation σ; also refer to Quesenberry (1993).
- Quesenberry (1991a, 1991b) introduced the so-called Q *chart* for short (or long) production runs, which standardizes and normalizes the data using probability integral transformations.

SAS examples illustrating these alternatives are provided in the SAS/QC sample library and are described by Rodriguez and Bynum (1992).

Analyzing the Difference from Nominal

The following example* is adapted from an application in aircraft component manufacturing. A metal extrusion process is used to make three slightly different models of the same component. The three product types (labeled M1, M2, and M3) are produced in small quantities because the process is expensive and time-consuming.

Figure 56.14 shows the structure of a SAS data set named OLD, which contains the diameter measurements for various short runs. Samples 1 to 30 are to be used to estimate the process standard deviation σ for the differences from nominal.

```
data old;
   input sample prodtype $ diameter;
datalines;
 1  M3  13.99
 2  M3  14.69

  ...

30  M3  14.35
;
run;
```

sample	prodtype	diameter
1	M3	13.99
2	M3	14.69
3	M3	13.86
4	M3	14.32
5	M3	13.23
6	M1	17.55
7	M1	14.26
8	M1	14.62
9	M1	12.97
10	M2	16.18
11	M2	15.29
12	M2	16.20
13	M3	13.89
14	M3	12.71
15	M3	14.32
16	M3	15.35
17	M2	15.08
18	M2	14.72
19	M2	14.79
20	M2	15.27
21	M2	15.95
22	M1	14.78
23	M1	15.19
24	M1	15.41
25	M1	16.26
26	M3	16.68
27	M3	15.60
28	M3	14.86
29	M3	16.67
30	M3	14.35

Figure 56.14. Diameter Measurements in the Data Set OLD

*Refer to Chapter 1 of Wheeler (1991a) for a similar example.

In short run applications involving many product types, it is common practice to maintain a database for the nominal values for the product types. Here, the nominal values are saved in a SAS data set named NOMVAL, which is listed in Figure 56.15.

prodtype	nominal
M1	15.0
M2	15.5
M3	14.8
M4	15.2

Figure 56.15. Nominal Values for Product Types in the Data Set NOMVAL

To compute the differences from nominal, you must merge the data with the nominal values. You can do this with the following SAS statements. Note that an IN= variable is used in the MERGE statement to allow for the fact that NOMVAL includes nominal values for product types that are not represented in OLD. Figure 56.16 lists the merged data set OLD.

```
proc sort data=old;
   by prodtype;
run;

data old;
   format diff 5.2 ;
   merge nomval old(in = a);
      by prodtype;
   if a;
   diff = diameter - nominal;
run;

proc sort data=old;
   by sample;
run;
```

Assume that the variability in the process is constant across product types. To estimate the common process standard deviation σ, you first estimate σ for each product type based on the average of the moving ranges of the differences from nominal. You can do this in several steps, the first of which is to sort the data and compute the average moving range with the SHEWHART procedure.

```
proc sort data=old;
   by prodtype;
run;

proc shewhart data=old;
   irchart diff*sample /
      nochart
      outlimits=baselim;
   by prodtype;
run;
```

sample	prodtype	diameter	nominal	diff
1	M3	13.99	14.8	-0.81
2	M3	14.69	14.8	-0.11
3	M3	13.86	14.8	-0.94
4	M3	14.32	14.8	-0.48
5	M3	13.23	14.8	-1.57
6	M1	17.55	15.0	2.55
7	M1	14.26	15.0	-0.74
8	M1	14.62	15.0	-0.38
9	M1	12.97	15.0	-2.03
10	M2	16.18	15.5	0.68
11	M2	15.29	15.5	-0.21
12	M2	16.20	15.5	0.70
13	M3	13.89	14.8	-0.91
14	M3	12.71	14.8	-2.09
15	M3	14.32	14.8	-0.48
16	M3	15.35	14.8	0.55
17	M2	15.08	15.5	-0.42
18	M2	14.72	15.5	-0.78
19	M2	14.79	15.5	-0.71
20	M2	15.27	15.5	-0.23
21	M2	15.95	15.5	0.45
22	M1	14.78	15.0	-0.22
23	M1	15.19	15.0	0.19
24	M1	15.41	15.0	0.41
25	M1	16.26	15.0	1.26
26	M3	16.68	14.8	1.88
27	M3	15.60	14.8	0.80
28	M3	14.86	14.8	0.06
29	M3	16.67	14.8	1.87
30	M3	14.35	14.8	-0.45

Figure 56.16. Data Merged with Nominal Values

The purpose of this procedure step is simply to save the average moving range for each product type in the OUTLIMITS= data set BASELIM, which is listed in Figure 56.17 (note that PRODTYPE is specified as a BY variable).

Control Limits By Product Type

prodtype	_VAR_	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_
M1	diff	sample	ESTIMATE	2	.002699796	3
M2	diff	sample	ESTIMATE	2	.002699796	3
M3	diff	sample	ESTIMATE	2	.002699796	3

LCLI	_MEAN_	_UCLI_	_LCLR_	_R_	_UCLR_	_STDDEV_
-3.13258	0.13000	3.39258	0	1.22714	4.00850	1.08753
-1.77795	-0.06500	1.64795	0	0.64429	2.10458	0.57098
-3.22641	-0.19143	2.84356	0	1.14154	3.72887	1.01166

Figure 56.17. Values of $\bar{R}$ by Product Type

To obtain a combined estimate of σ, you can use the MEANS procedure to average the average ranges in BASELIM and then divide by the unbiasing constant d_2.

```
proc means data=baselim noprint;
   var _r_;
   output out=difflim (keep=_r_) mean=_r_;
run;

data difflim;
   set difflim;
   drop _r_;
   length _var_ _subgrp_ $ 8;
   _var_    = 'diff';
   _subgrp_ = 'sample';
   _mean_   = 0.0;
   _stddev_ = _r_ / d2(2);
   _limitn_ = 2;
   _sigmas_ = 3;
run;
```

The data set DIFFLIM is structured for subsequent use by the SHEWHART procedure as an input LIMITS= data set. The variables in a LIMITS= data set provide pre-computed control limits or—as in this case—the parameters from which control limits are to be computed. These variables have reserved names that begin and end with the underscore character. Here, the variable _STDDEV_ saves the estimate of σ, and the variable _MEAN_ saves the mean of the differences from nominal. Recall that this mean is zero, since the nominal values are assumed to represent the process mean for each product type. The identifier variables _VAR_ and _SUBGRP_ record the names of the process and subgroup variables (these variables are critical in applications involving many product types). The variable _LIMITN_ is assigned a value of 2 to specify moving ranges of two consecutive measurements, and the variable

SIGMAS is assigned a value of 3 to specify 3σ limits. The data set DIFFLIM is listed in Figure 56.18.

Control Limit Parameters For Differences

var	_subgrp_	_mean_	_stddev_	_limitn_	_sigmas_
diff	sample	0	0.89006	2	3

Figure 56.18. Estimates of Mean and Standard Deviation

Now that the control limit parameters are saved in DIFFLIM, diameters for an additional 30 parts (samples 31 to 60) are measured and saved in a SAS data set named NEW. You can construct short run control charts for this data by merging the measurements in NEW with the corresponding nominal values in NOMVAL, computing the differences from nominal, and then constructing the short run individual measurements and moving range charts.

```
proc sort data=new;
   by prodtype;
run;

data new;
   format diff 5.2 ;
   merge nomval new(in = a);
      by prodtype;
   if a;
   diff = diameter - nominal;
   label sample   = 'Sample Number'
         prodtype = 'Model';
run;

proc sort data=new;
   by sample;
run;

symbol1 v=dot    c=black;
symbol2 v=plus   c=black;
symbol3 v=circle c=black;
title 'Chart for Difference from Nominal';
proc shewhart data=new limits=difflim;
   irchart diff*sample=prodtype / split    = '/';
   label diff = 'Difference/Moving Range';
run;
```

The chart is displayed in Figure 56.19. Note that the product types are identified with symbol markers as requested by specifying PRODTYPE as a *symbol-variable*.

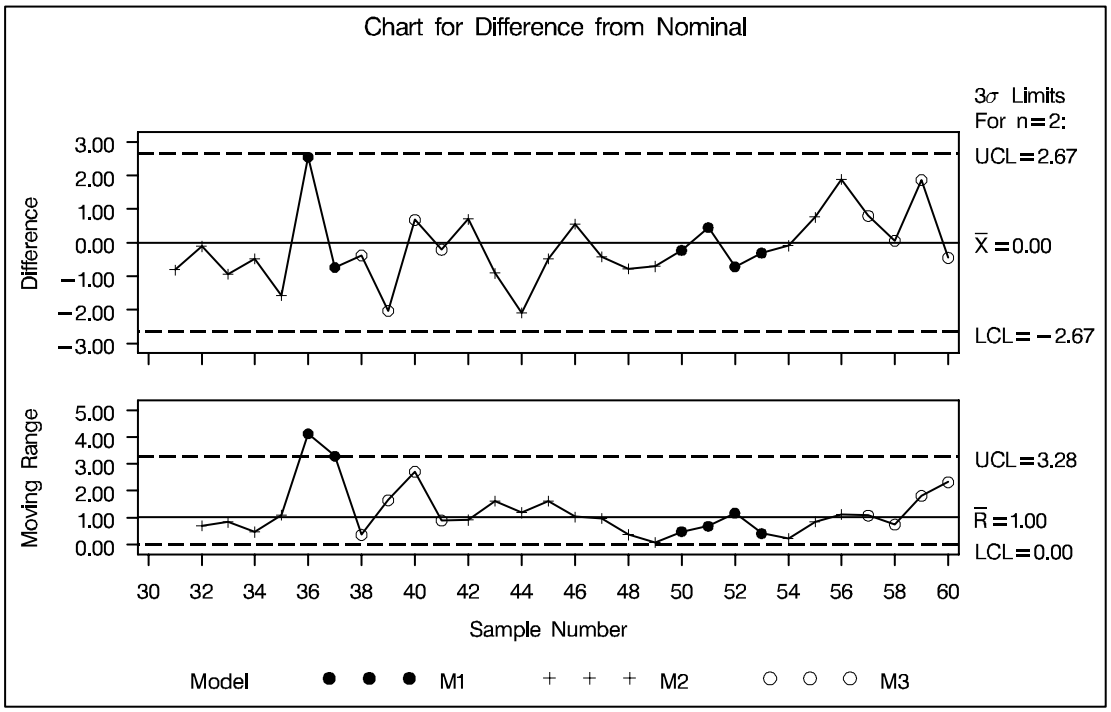


Figure 56.19. Short Run Control Chart

You can also identify the product types with a legend by specifying PRODTYPE as a _PHASE_ variable.

```
title 'Chart for Difference from Nominal';
proc shewhart data=new (rename=(prodtype=_phase_)) limits=difflim;
   irchart diff*sample /
      readphases = all
      phaseref
      phasebreak
      phaselegend
      split      = '/';
   label diff = 'Difference/Moving Range';
run;
```

The display is shown in Figure 56.20. Note that the PHASEBREAK option is used to suppress the connection of adjacent points in different phases (product types).

In some applications, it may be useful to replace the moving range chart with a plot of the nominal values. You can do this with the TRENDVAR= option in the XCHART statement* provided that you reset the value of _LIMITN_ to 1 to specify a subgroup sample of size one.

```
data difflim;
   set difflim;
   _var_    = 'diameter';
   _limitn_ = 1;
run;
```

*The TRENDVAR= option is not available in the IRCHART statement.

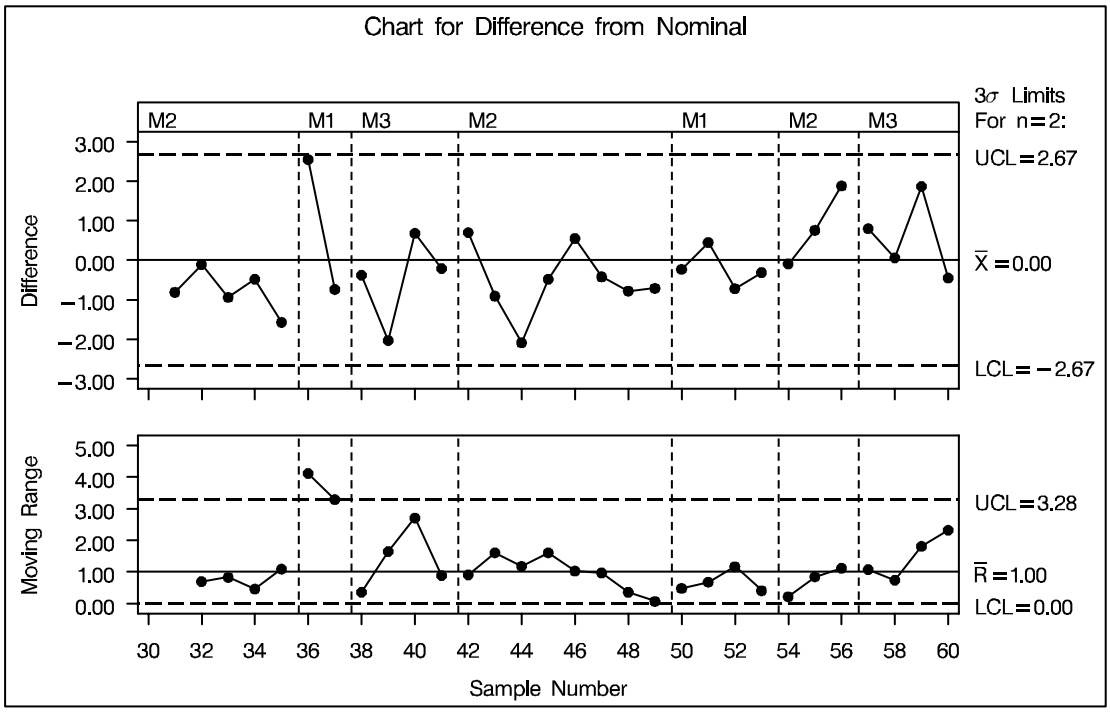


Figure 56.20. Identification of Product Types

```
title 'Differences and Nominal Values';
proc shewhart data=new limits=difflim;
   xchart diameter*sample (prodtype) /
      nolimitslegend
      nolegend
      split          = '//'
      blockpos       = 3
      blocklabtype   = scaled
      blocklabelpos  = left
      xsymbol        = xbar
      trendvar       = nominal;
   label diameter    = 'Difference/Nominal'
         prodtype    = 'Product';
run;
```

The display is shown in Figure 56.21. Note that you identify the product types by specifying PRODTYPE as a *block variable* enclosed in parentheses after the subgroup variable SAMPLE. The BLOCKLABTYPE= option specifies that values of the block variable are to be scaled (if necessary) to fit the space available in the block legend. The BLOCKLABELPOS= option specifies that the label of the block variable is to be displayed to the left of the block legend.

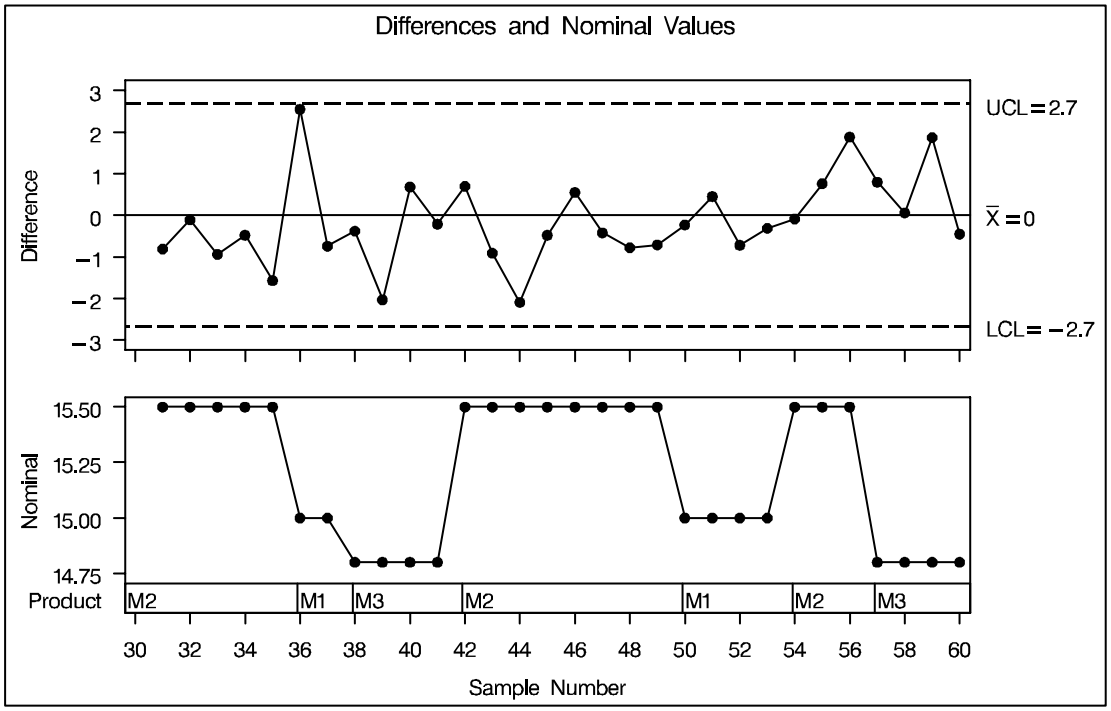


Figure 56.21. Short Run Control Chart with Nominal Values

Testing for Constant Variances

The difference-from-nominal chart should be accompanied by a test that checks whether the variances for each product type are identical (homogeneous). Levene's test of homogeneity is particularly appropriate for short run applications because it is robust to departures from normality; refer to Snedecor and Cochran (1980). You can implement Levene's method by using the GLM procedure to construct a one-way analysis of variance for the absolute deviations of the diameters from averages within product types.

```
proc sort data=old;
   by prodtype;
run;

proc means data=old noprint;
   var diameter;
   by prodtype;
   output out=oldmean (keep=prodtype diammean) mean=diammean;
run;

data old;
   merge old oldmean;
      by prodtype;
   absdev = abs( diameter - diammean );
run;

proc means data=old noprint;
  var absdev;
  by prodtype;
  output out=stats n=n mean=mean css=css std=std;
run;
```

A partial listing of the results is displayed in Figure 56.22. The large p-value (0.3386) indicates that the data do not reject the hypothesis of homogeneity.

The GLM Procedure

Dependent Variable: absdev

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	1.02901063	0.51450532	1.13	0.3373
Error	27	12.27381243	0.45458565		
Corrected Total	29	13.30282306			

Figure 56.22. Levene's Test of Variance Homogeneity

Standardizing Differences from Nominal

When the variances across product types are *not* constant, various authors recommend standardizing the differences from nominal and displaying them on a common chart with control limits at ± 3.

To illustrate this method, assume that the hypothesis of homogeneity is rejected for the differences in OLD. Then you can use the product-specific estimates of σ in BASELIM to standardize the differences from nominal in NEW and create the standardized chart as follows:

```
proc sort data=new;
   by prodtype;
run;

data new;
   keep sample prodtype z diff diameter nominal _stddev_;
   label sample = 'Sample Number';
   format diff 5.2 ;
   merge baselim new(in = a);
      by prodtype;
   if a;
   z = (diameter - nominal) / _stddev_ ;
run;

proc sort data=new;
   by sample;
run;
```

```
title 'Standardized Chart';
proc shewhart data=new;
   irchart z*sample (prodtype) /
      blocklabtype  = scaled
      mu0           = 0
      sigma0        = 1
      split         = '/';
   label prodtype = 'Product Classification'
         z = 'Standardized Difference/Moving Range';
run;
```

Note that the options MU0= and SIGMA= specify that the control limits for the standardized differences from nominal are to be based on the parameters $\mu = 0$ and $\sigma = 1$. The chart is displayed in Figure 56.23.

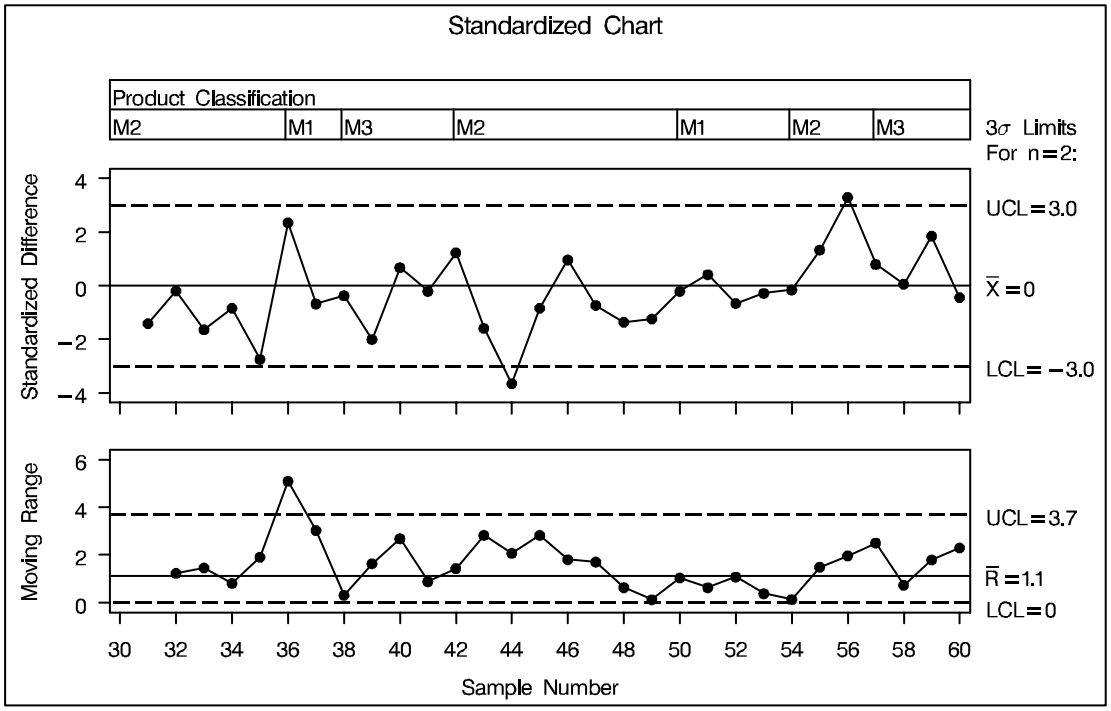


Figure 56.23. Standardized Difference Chart

Nonnormal Process Data

See SHWNONN in the SAS/QC Sample Library

A number of authors have pointed out that Shewhart charts for subgroup means work well whether the measurements are normally distributed or not.* On the other hand, the interpretation of standard control charts for individual measurements (X charts) is affected by departures from normality.

In situations involving a large number of measurements, it may be possible to subgroup the data and construct an $\bar{X}$ chart instead of an X chart. However, the mea-

*Refer to Schilling and Nelson (1976) and Wheeler (1991b).

surements should not be subgrouped arbitrarily for this purpose.* If subgrouping is not possible, two alternatives are to transform the data to normality (preferably with a simple transformation such as the log transformation) or modify the usual limits based on a suitable model for the data distribution.

The second of these alternatives is illustrated here with data from a study conducted by a service center. The time taken by staff members to answer the phone was measured, and the delays were saved as values of a variable named TIME in a SAS data set named CALLS. A partial listing of CALLS is shown in Figure 56.24.

Creating a Preliminary Individual Measurements Chart

As a first step, the delays were analyzed using an X chart created with the following statements. The chart is displayed in Figure 56.25.

```
title 'Standard Analysis of Individual Delays';
proc shewhart data=calls;
   irchart time * recnum /
      rtmplot   = schematic
      outlimits = delaylim
      cboxfill  = gray
      nochart2 ;
   label recnum = 'Record Number'
         time   = 'Delay in minutes' ;
run;
```

You may be inclined to conclude that the 41st point signals a special cause of variation. However, the box plot in the right margin (requested with the RTMPLOT= option) indicates that the distribution of delays is skewed. Thus, the reason that the measurements are grouped well within the control limits is that the limits are incorrect and not that the process is too good for the limits.

Note: This example assumes the process is in statistical control; otherwise, the box plot could not be interpreted as a representation of the process distribution. You can check the assumption of normality with goodness-of-fit tests by using the CAPABILITY procedure, as shown in the statements that follow.

*Refer to Wheeler and Chambers (1986) for a discussion of subgrouping.

recnum	time
1	3.233
2	3.110
3	3.136
4	2.899
5	2.838
6	2.459
7	3.716
8	2.740
9	2.487
10	2.635
11	2.676
12	2.905
13	3.431
14	2.663
15	3.437
16	2.823
17	2.596
18	2.633
19	3.235
20	2.701
21	3.202
22	2.725
23	3.151
24	2.464
25	2.662
26	3.188
27	2.640
28	2.541
29	3.033
30	2.993
31	2.636
32	2.481
33	3.191
34	2.662
35	2.967
36	3.300
37	2.530
38	2.777
39	3.353
40	3.614
41	4.288
42	2.442
43	2.552
44	2.613
45	2.731
46	2.780
47	3.588
48	2.612
49	2.579
50	2.871

Figure 56.24. Answering Times from the Data Set CALLS

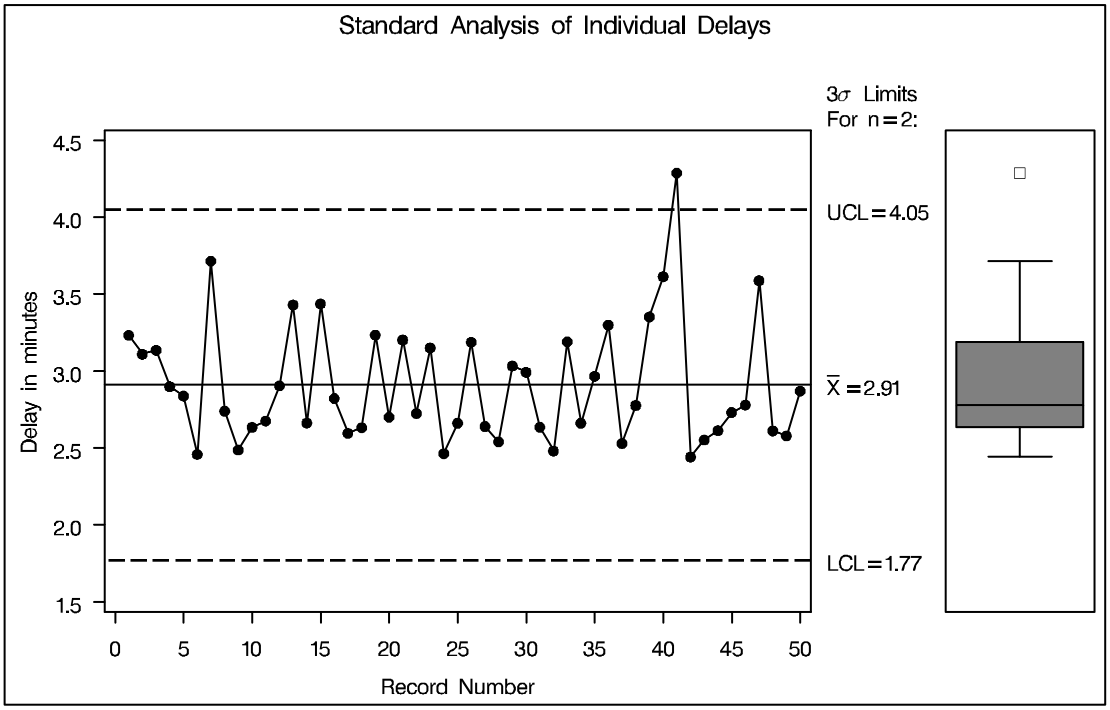


Figure 56.25. Standard Control Limits for Delays

Calculating Probability Limits

The OUTLIMITS= option saves the control limits from the chart in Figure 56.25 in a SAS data set named DELAYLIM, which is listed in Figure 56.26.

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLI_	_MEAN_	_UCLI_	_STDDEV_
time	recnum	ESTIMATE	2	.002699796	3	1.77008	2.91038	4.05068	0.38010

Figure 56.26. Control Limits for Standard Chart from the Data Set CALLS

The control limits can be replaced with the corresponding percentiles from a fitted lognormal distribution. The equation for the lognormal density function is

$$f(x) = \frac{1}{x\sqrt{2\pi}\sigma} \exp\left(-\frac{(\log(x)-\zeta)^2}{2\sigma^2}\right) \quad x > 0$$

where σ denotes the shape parameter and ζ denotes the scale parameter.

The following statements use the CAPABILITY procedure to fit a lognormal model and superimpose the fitted density on a histogram of the data, shown in Figure 56.27:

```
title 'Lognormal Fit for Delay Distribution';
proc capability data=calls noprint;
   histogram time /
      lognormal(threshold=2.3 color=black w=2)
      outfit = lnfit
      nolegend ;
   inset n = 'Number of Calls'
      lognormal( sigma = 'Shape' (4.2)
                 zeta  = 'Scale' (5.2)
                 theta ) / pos    = ne;
   label time  = 'Answering Delay (minutes)';
run;
```

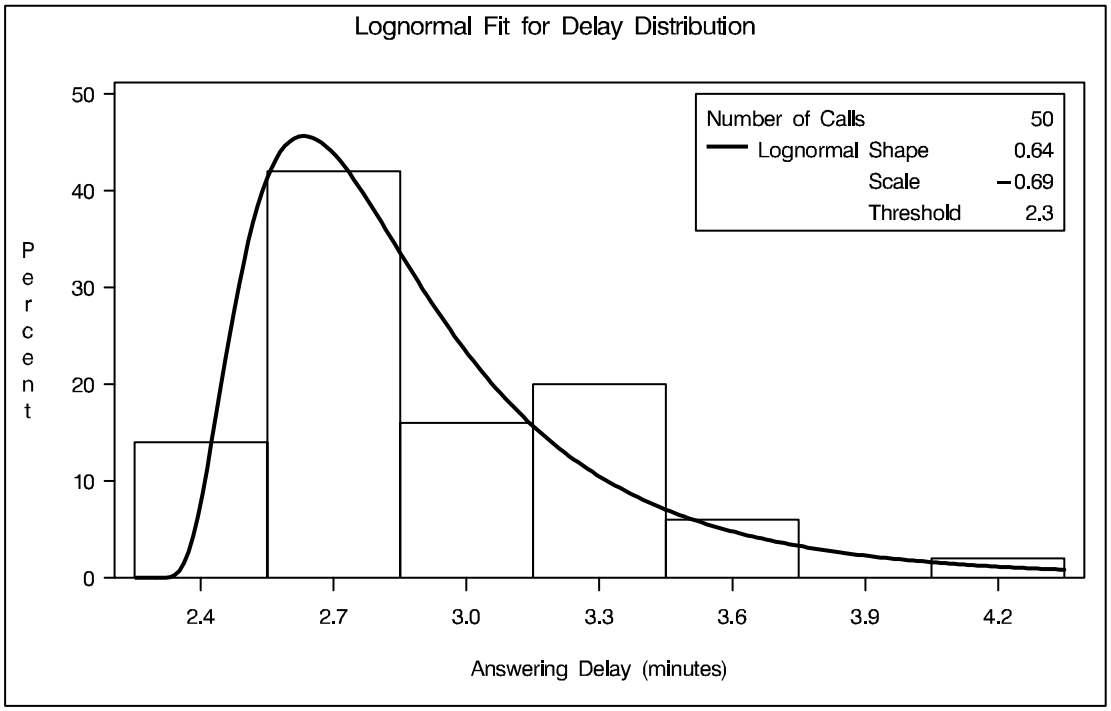


Figure 56.27. Distribution of Delays

Parameters of the fitted distribution and results of goodness-of-fit tests are saved in the data set LNFIT, which is listed in Figure 56.28. The large p-values for the goodness-of-fit tests are evidence that the lognormal model provides a good fit.

VAR	_CURVE_	_LOCATN_	_SCALE_	_SHAPE1_	_MIDPTN_
time	LNORMAL	2.3	-0.68910	0.64110	4.2

ADASQ	_ADP_	_CVMWSQ_	_CVMP_	_KSD_	_KSP_
0.34854	0.47465	0.058737	0.40952	0.092223	0.15

Figure 56.28. Parameters of Fitted Lognormal Model in the Data Set LNFIT

The following statements replace the control limits in DELAYLIM with limits computed from percentiles of the fitted lognormal model. The $100\alpha^{\text{th}}$ percentile of the lognormal distribution is $P_\alpha = \exp(\sigma\Phi^{-1}(\alpha) + \zeta)$, where Φ^{-1} denotes the inverse standard normal cumulative distribution function. The SHEWHART procedure constructs an X chart with the modified limits, displayed in Figure 56.29.

```
data delaylim;
   merge delaylim lnfit;
   drop _sigmas_ ;
   _lcli_ = _locatn_ + exp(_scale_+probit(0.5*_alpha_)*_shape1_);
   _ucli_ = _locatn_ + exp(_scale_+probit(1-.5*_alpha_)*_shape1_);
   _mean_ = _locatn_ + exp(_scale_+0.5*_shape1_*_shape1_);
run;

title 'Lognormal Control Limits for Delays';
proc shewhart data=calls limits=delaylim;
   irchart time*recnum /
      rtmplot   = schematic
      cboxfill  = gray
      nochart2 ;
   label recnum = 'Record Number'
         time   = 'Delay in minutes' ;
run;
```

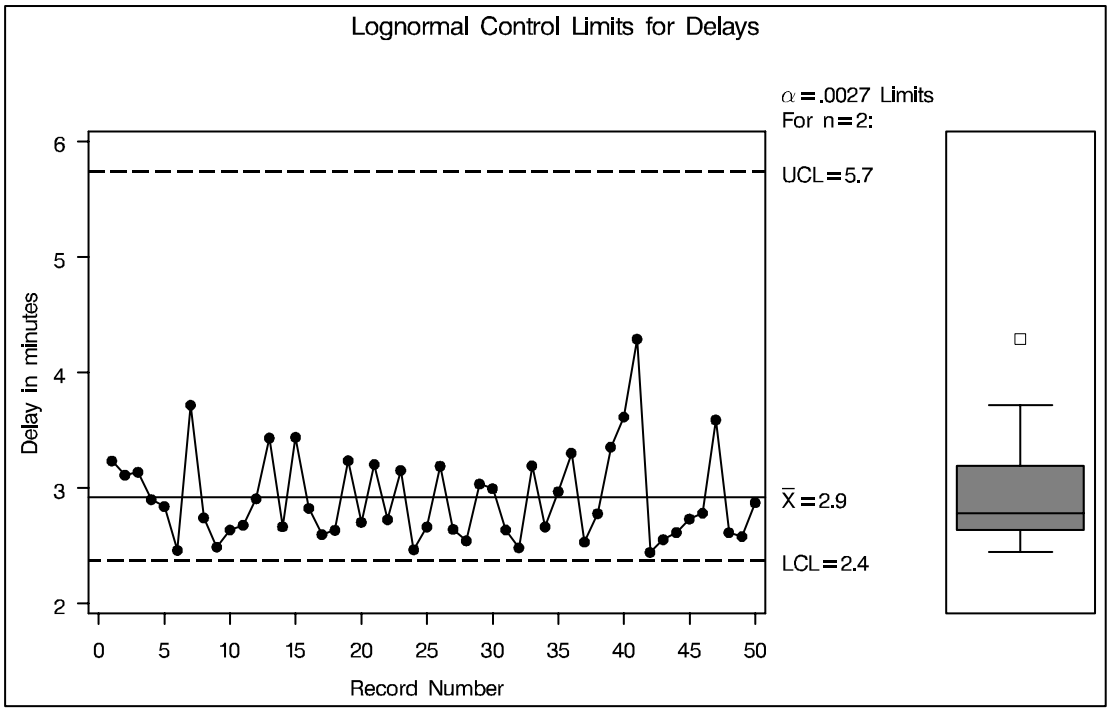


Figure 56.29. Adjusted Control Limits for Delays

Clearly the process is in control, and the control limits (particularly the lower limit) are appropriate for the data. The particular probability level $\alpha = 0.0027$ associated with these limits is somewhat immaterial, and other values of α such as 0.001 or 0.01 could be specified with the ALPHA= option in the original IRCHART statement.

Multivariate Control Charts

See SHWT2 in the SAS/QC Sample Library

In many industrial applications, the output of a process characterized by p variables that are measured simultaneously. Independent variables can be charted individually, but if the variables are correlated, a multivariate chart is needed to determine whether the process is in control.

Many types of multivariate control charts have been proposed; refer to Alt (1985) for an overview. Denote the i^{th} measurement on the j^{th} variable as X_{ij} for $i = 1, 2, \ldots, n$, where n is the number of measurements, and $j = 1, 2, \ldots, p$. Standard practice is to construct a chart for a statistic T_i^2 of the form

$$T_i^2 = (\mathbf{X}_i - \bar{\mathbf{X}}_n)' \mathbf{S}_n^{-1} (\mathbf{X}_i - \bar{\mathbf{X}}_n)$$

where

$$\bar{X}_j = \tfrac{1}{n} \textstyle\sum_{i=1}^{n} X_{ij} \quad , \quad \mathbf{X}_i = \begin{bmatrix} X_{i1} \\ X_{i2} \\ \vdots \\ X_{ip} \end{bmatrix}, \quad \bar{\mathbf{X}}_n = \begin{bmatrix} \bar{X}_1 \\ \bar{X}_2 \\ \vdots \\ \bar{X}_p \end{bmatrix}$$

and

$$\mathbf{S}_n = \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{X}_i - \bar{\mathbf{X}}_n)(\mathbf{X}_i - \bar{\mathbf{X}}_n)'$$

It is assumed that $\mathbf{X}_i$ has a p-dimensional multivariate normal distribution with mean vector $\mu = (\mu_1 \mu_2 \cdots \mu_p)'$ and covariance matrix $\mathbf{\Sigma}$ for $i = 1, 2, \ldots, n$. Depending on the assumptions made about the parameters, a χ^2, Hotelling T^2, or beta distribution is used for T_i^2, and the percentiles of this distribution yield the control limits for the multivariate chart.

In this example, a multivariate control chart is constructed using a beta distribution for T_i^2. The beta distribution is appropriate when the data are individual measurements (rather than subgrouped measurements) and when μ and $\mathbf{\Sigma}$ are estimated from the data being charted. In other words, this example illustrates a start-up phase chart where the control limits are determined from the data being charted.

Calculating the Chart Statistic

In this situation, it was shown by Gnanadesikan and Kettenring (1972), using a result of Wilks (1962), that T_i^2 is exactly distributed as a multiple of a variable with a beta distribution. Specifically,

$$T_i^2 \sim \frac{(n-1)^2}{n} B\left(\frac{p}{2}, \frac{n-p-1}{2}\right)$$

Tracy, Young, and Mason (1992) used this result to derive initial control limits for a multivariate chart based on three quality measures from a chemical process in the start-up phase: percent of impurities, temperature, and concentration. The remainder of this section describes the construction of a multivariate control chart using their data, which are given here by the data set STARTUP.

```
data startup;
   input sample impure temp conc;
   label sample = 'Sample Number'
         impure = 'Impurities'
         temp   = 'Temperature'
         conc   = 'Concentration' ;
   datalines;
 1  14.92  85.77  42.26
 2  16.90  83.77  43.44
 3  17.38  84.46  42.74
 4  16.90  86.27  43.60
 5  16.92  85.23  43.18
 6  16.71  83.81  43.72
 7  17.07  86.08  43.33
 8  16.93  85.85  43.41
 9  16.71  85.73  43.28
10  16.88  86.27  42.59
11  16.73  83.46  44.00
12  17.07  85.81  42.78
13  17.60  85.92  43.11
14  16.90  84.23  43.48
;
run;
```

In preparation for the computation of the control limits, the sample size is calculated and parameter variables are defined.

```
proc means data=startup noprint ;
   var impure temp conc;
   output out=means n=n;
run;

data startup;
   if _n_ = 1 then set means;
   set startup;
   p        = 3;
   _subn_   = 1;
   _limitn_ = 1;
run;
```

Next, the PRINCOMP procedure is used to compute the principal components of the variables and save them in an output data set named PRIN.

```
proc princomp data=startup out=prin outstat=scores std cov;
   var impure temp conc;
run;
```

The following statements compute T_i^2 and its exact control limits, using the fact that T_i^2 is the sum of squares of the principal components.* Note that these statements

*Refer to Jackson (1980).

create several special SAS variables so that the data set PRIN can subsequently be read as a TABLE= input data set by the SHEWHART procedure. These special variables begin and end with an underscore character. The data set PRIN is listed in Figure 56.30.

```
data prin (rename=(tsquare=_subx_));
   length _var_ $ 8 ;
   drop prin1 prin2 prin3 _type_ _freq_;
   set prin;
   comp1   = prin1*prin1;
   comp2   = prin2*prin2;
   comp3   = prin3*prin3;
   tsquare = comp1 + comp2 + comp3;
   _var_   = 'tsquare';
   _alpha_ = 0.05;
   _lclx_  = ((n-1)*(n-1)/n)*betainv(_alpha_/2, p/2, (n-p-1)/2);
   _mean_  = ((n-1)*(n-1)/n)*betainv(0.5, p/2, (n-p-1)/2);
   _uclx_  = ((n-1)*(n-1)/n)*betainv(1-_alpha_/2, p/2, (n-p-1)/2);
   label tsquare = 'T Squared'
         comp1   = 'Comp 1'
         comp2   = 'Comp 2'
         comp3   = 'Comp 3';
run;
```

T2 Chart For Chemical Example

var	n	sample	impure	temp	conc	p	_subn_	_limitn_	comp1
tsquare	14	1	14.92	85.77	42.26	3	1	1	0.79603
tsquare	14	2	16.90	83.77	43.44	3	1	1	1.84804
tsquare	14	3	17.38	84.46	42.74	3	1	1	0.33397
tsquare	14	4	16.90	86.27	43.60	3	1	1	0.77286
tsquare	14	5	16.92	85.23	43.18	3	1	1	0.00147
tsquare	14	6	16.71	83.81	43.72	3	1	1	1.91534
tsquare	14	7	17.07	86.08	43.33	3	1	1	0.58596
tsquare	14	8	16.93	85.85	43.41	3	1	1	0.29543
tsquare	14	9	16.71	85.73	43.28	3	1	1	0.23166
tsquare	14	10	16.88	86.27	42.59	3	1	1	1.30518
tsquare	14	11	16.73	83.46	44.00	3	1	1	3.15791
tsquare	14	12	17.07	85.81	42.78	3	1	1	0.43819
tsquare	14	13	17.60	85.92	43.11	3	1	1	0.41494
tsquare	14	14	16.90	84.23	43.48	3	1	1	0.90302

comp2	comp3	_subx_	_alpha_	_lclx_	_mean_	_uclx_
10.1137	0.01606	10.9257	0.05	0.24604	2.44144	7.13966
0.0162	0.17681	2.0410	0.05	0.24604	2.44144	7.13966
0.1538	5.09491	5.5827	0.05	0.24604	2.44144	7.13966
0.3289	2.76215	3.8640	0.05	0.24604	2.44144	7.13966
0.0165	0.01919	0.0372	0.05	0.24604	2.44144	7.13966
0.0645	0.27362	2.2534	0.05	0.24604	2.44144	7.13966
0.4079	0.44146	1.4354	0.05	0.24604	2.44144	7.13966
0.1729	0.73939	1.2077	0.05	0.24604	2.44144	7.13966
0.0001	0.44483	0.6766	0.05	0.24604	2.44144	7.13966
0.0004	0.86364	2.1692	0.05	0.24604	2.44144	7.13966
0.0274	0.98639	4.1717	0.05	0.24604	2.44144	7.13966
0.0823	0.87976	1.4003	0.05	0.24604	2.44144	7.13966
1.6153	0.30167	2.3320	0.05	0.24604	2.44144	7.13966
0.0001	0.00010	0.9032	0.05	0.24604	2.44144	7.13966

Figure 56.30. The Data Set PRIN

You can now use the data set PRIN as input to the SHEWHART procedure to create the multivariate control chart displayed in Figure 56.31.

```
title 'T' m=(+0,+0.5) '2'
          m=(+0,-0.5) ' Chart For Chemical Example';
proc shewhart table=prin;
   xchart tsquare*sample /
      xsymbol  = mu
      nolegend ;
run;
```

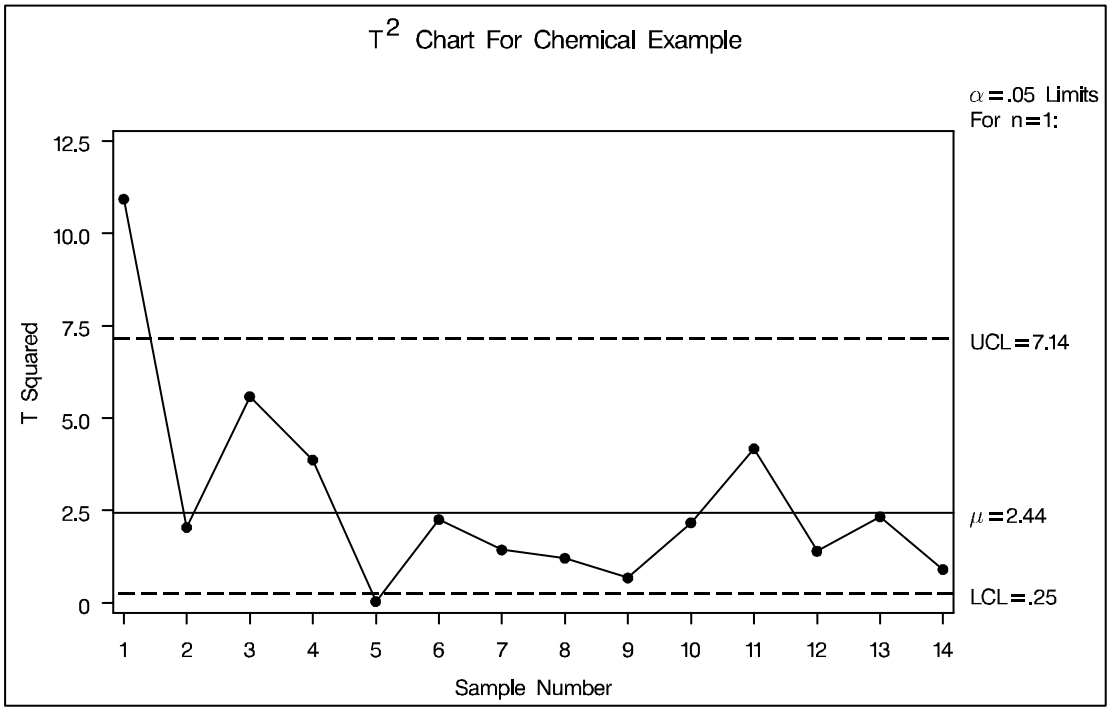


Figure 56.31. Multivariate Control Chart for Chemical Process

The methods used in this example easily generalize to other types of multivariate control charts. You can create charts using the χ^2 and F distributions by using the appropriate CINV or FINV function in place of the BETAINV function in the statements on page 2035. For details, refer to Alt (1985), Jackson (1980, 1991), and Ryan (1989).

Examining the Principal Component Contributions

You can use the *star options* in the SHEWHART procedure to superimpose points on the chart with stars whose vertices represent standardized values of the squares of the three principal components used to determine T_i^2.

```
title 'T' m=(+0,+0.5) '2'
          m=(+0,-0.5) ' Chart For Chemical Example';
symbol value=none;
proc shewhart table=prin;
   xchart tsquare*sample /
      starvertices  = (comp1 comp2 comp3)
      startype      = wedge
      cstars        = black
      starlegend    = none
      starlabel     = first
      staroutradius = 4
      npanelpos     = 14
      xsymbol       = mu
      nolegend ;
run;
```

The chart is displayed in Figure 56.32. In situations where the principal components have a physical interpretation, the star chart can be a helpful diagnostic for determining the relative contributions of the different components.

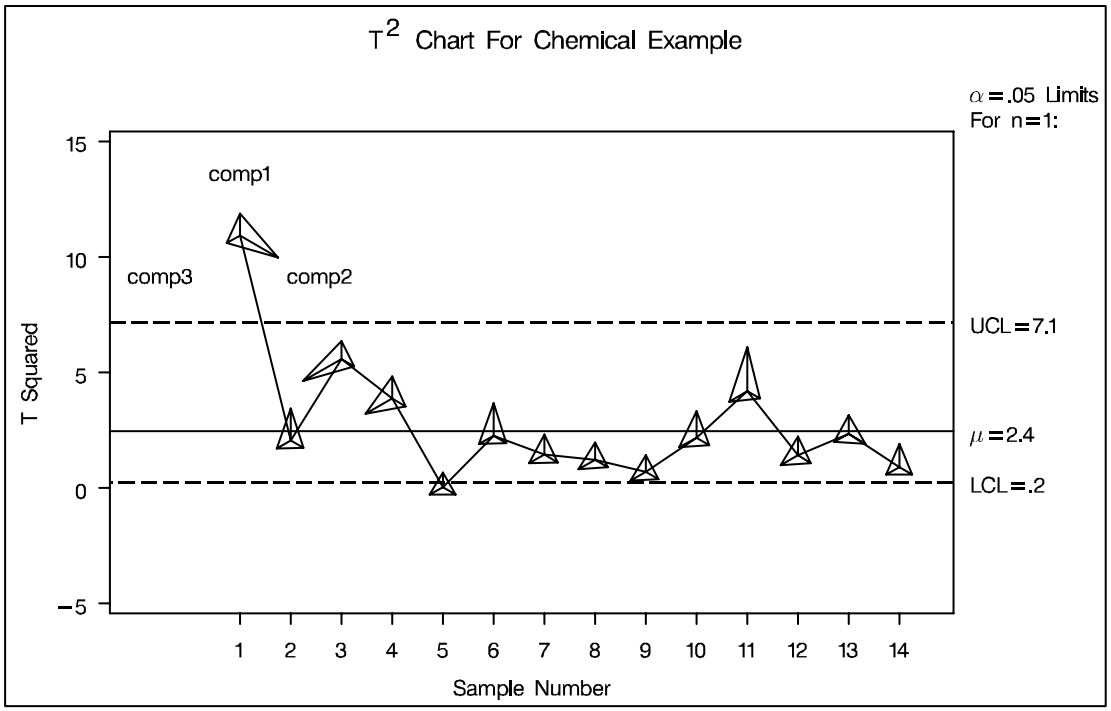


Figure 56.32. Multivariate Control Chart Displaying Principal Components

For more information about star charts, see "Displaying Auxiliary Data with Stars" on page 1948, or consult the entries for the STARVERTICES= and related options in Chapter 53, "Dictionary of Options."

Principal components are not the only approach that can be used to interpret multivariate control charts. This problem has recently been studied by a number of authors, including Doganaksoy and others (1991), Hawkins (1991, 1993), and Mason and others (1993).

Chapter 57
Interactive Control Charts

Chapter Contents

OVERVIEW 2041

DETAILS 2042
Saving Graphics Coordinates in a Control Chart 2042
Associating URLs with Subgroups in HTML 2044
URLS and Tests for Special Causes 2046

Chapter 57
Interactive Control Charts

Overview

This chapter describes two approaches for creating an interactive control chart which allows an end user to "drill down" into subgroup data points and display information not contained in the chart itself. For example, the end user might want to be able to click on a subgroup to

- list the individual measurements in the subgroup
- diagnose an out-of-control point by viewing a Pareto chart of the most common problems affecting the process
- view a list of recommended corrective actions
- trace the raw materials used to manufacture a batch of product

The two approaches for creating interactive control charts are as follows:

- saving graphics coordinate data from control charts for use in creating SAS/AF applications
- associating Uniform Resource Locators (URLs) with subgroups to produce "clickable" control charts in HTML

The options described in this chapter can be specified in all the chart statements available in the SHEWHART procedure.

Details

Saving Graphics Coordinates in a Control Chart

You can specify an WEBOUT= data set in any chart statement to save graphics coordinate information for a control chart. The WEBOUT= data set is an extension of the OUTTABLE= data set, which contains the subgroup summary statistics, control limits and related information found in an OUTTABLE= data set, as well as coordinate data. The additional coordinate variables are listed in Table 57.1.

Table 57.1. WEBOUT= Data Set

Variable	Description
_X1_	x-coordinate of lower left corner of primary chart subgroup bounding box
_Y1_	y-coordinate of lower left corner of primary chart subgroup bounding box
_X2_	x-coordinate of upper right corner of primary chart subgroup bounding box
_Y2_	y-coordinate of upper right corner of primary chart subgroup bounding box
_X*n*_	x-coordinate for point *n* of the subgroup shape
_Y*n*_	y-coordinate for point *n* of the subgroup shape
_X1_2_	x-coordinate of lower left corner of secondary chart subgroup bounding box
_Y1_2_	y-coordinate of lower left corner of secondary chart subgroup bounding box
_X2_2_	x-coordinate of upper right corner of secondary chart subgroup bounding box
_Y2_2_	y-coordinate of upper right corner of secondary chart subgroup bounding box
SHAPE	shape of primary chart subgroup bounding area
NXY	number of points defining primary chart subgroup bounding area
GRAPH	name of primary chart graphics entry
GRAPH2	name of secondary chart graphics entry
DXMIN	value of lowest major tick mark on horizontal axis
DXMAX	value of highest major tick mark on horizontal axis
XMIN	x-coordinate of lowest major tick mark on horizontal axis
XMAX	x-coordinate of highest major tick mark on horizontal axis
DYMIN	value of lowest major tick mark on vertical axis
DYMAX	value of highest major tick mark on vertical axis
YMIN	y-coordinate of lowest major tick mark on vertical axis
YMAX	y-coordinate of highest major tick mark on vertical axis
XMIN2	x-coordinate of lowest major tick mark on secondary chart horizontal axis
XMAX2	x-coordinate of highest major tick mark on secondary chart horizontal axis
DYMIN2	value of lowest major tick mark on secondary chart vertical axis
DYMAX2	value of highest major tick mark on secondary chart vertical axis
YMIN2	y-coordinate of lowest major tick mark on secondary chart vertical axis
YMAX2	y-coordinate of highest major tick mark on secondary chart vertical axis

You can use the coordinate data saved in the WEBOUT= data set to create a "clickable" control chart in a SAS/AF application. The variables _X1_, _Y1_, _X2_ and _Y2_ contain the coordinates of the lower left and upper right corners of a rectangular *bounding box* associated with each subgroup on the primary chart. This box defines the clickable area associated with the subgroup when the chart is incorporated into a SAS/AF application. It contains the symbol used to plot the subgroup data, or the junction of line segments representing the subgroup if no plotting symbol is used. The variables _X1_2_, _Y1_2_, _X2_2_ and _Y2_2_ contain coordinates of the corners of subgroup bounding boxes for a secondary chart.

If you use the BOXCHART statement, each subgroup is represented by a box-and-whisker plot rather than a single symbol. The subgroup's bounding box is defined by the sides of the box-and-whisker plot and its lower and upper quartiles, regardless of the BOXSTYLE= value in effect.

If you specify the STARVERTICES= option, each subgroup is represented by a polygon or star with a vertex corresponding to each of the STARVERTICES= variables. The clickable area for a subgroup is the polygon with these vertices, regardless of the STARTYPE= value specified. In the WEBOUT= data set the value of the _SHAPE_ variable is POLY and the _NXY_ variable contains the number of vertices in the polygon. The variables _Xn_ and _Yn_, where $n = 1$ to the value of _NXY_, contain the coordinates of the vertices of a subgroup's polygon. When the STARVERTICES= option is not used, the value of _SHAPE_ is always RECT and the value of _NXY_ is always 2.

When a control chart spans multiple panels (pages), the panels reside in separate SAS graphics entries. The _GRAPH_ character variable records the name of the graphics entry containing the panel on which a given subgroup is plotted. This is the same name that appears in the PROC GREPLAY menu. When the SEPARATE option is used, primary and secondary charts are displayed on different graphics entries. The _GRAPH2_ variable records the name of the graphics entry containing the secondary chart panel where a subgroup appears. When the SEPARATE option is not used, the values of _GRAPH_ and _GRAPH2_ will be the same for a given subgroup.

The variables _DXMIN_, _DXMAX_, _XMIN_ and _XMAX_ provide the data values and graphics coordinates associated with the lowest and highest major tick marks on the horizontal (subgroup) axis. The variables _DYMIN_, _DYMAX_, _YMIN_ and _YMAX_ provide the analogous values for the vertical axis. Through a simple linear transformation in your SAS/AF application you can use this information to convert from percent screen units to "data" units and vice versa.

The variables _XMIN2_ and _XMAX2_ contain the graphics coordinates associated with the lowest and highest major tick marks on the horizontal axis of a secondary chart. No variables for the corresponding data values are required, since they are always identical to those for the primary chart.

The variables _DYMIN2_, _DYMAX2_, _YMIN2_ and _YMAX2_ contain the data and coordinate values for the lowest and highest tick marks on the vertical axis of a secondary chart. A SAS/AF program receives the (x,y) coordinates for the location of the cursor when the user clicks on a subgroup data point. The application can

determine whether (x,y) lies within any of the boxes whose coordinates are saved in the WEBOUT= data set. If so, the program can determine which subgroup was selected on the primary or secondary chart and can check the _TESTS_ and _TESTS2_ variables included in the WEBOUT= data set to determine whether an out-of-control condition has been signaled.

Notes:

1. Graphics coordinates are scaled in percent screen units from 0 to 100, where (0,0) represents the lower-left corner of the screen and (100,100) represents the upper-right corner of the screen. Because SAS/AF applications define the origin of the vertical axis at the top of the screen, it will be necessary to subtract the y-coordinates from 100 in your SCL program.
2. The variables _X1_2_, _Y1_2_, _X2_2_, _Y2_2_, _GRAPH2_, _XMIN2_, _XMAX2_, _YMIN2_, _YMAX2_, _DYMIN2_ and _DYMAX2_ appear in the WEBOUT= data set only when a secondary chart is produced. A secondary chart is produced by the IRCHART, MRCHART, XRCHART and XSCHART statements and by the BOXCHART, MCHART and XCHART statements when the TRENDVAR= option is specified.
3. When the subgroup variable is a character variable, the value of _DXMIN_ is zero and the value of _DXMAX_ is the number of subgroups in the input data set minus one.
4. A bounding box circumscribes a point displayed on a chart and its dimensions depend on the size of the symbol marker used to display the point. If no symbol marker is specified, a small default size is used for the box. If a large number of subgroups are displayed on a panel, the subgroup symbols may overlap, so it is possible for a user to inadvertently select more than one point.

Associating URLs with Subgroups in HTML

You can use the Output Delivery System (ODS) to produce an HTML file containing a control chart created by the SHEWHART procedure. The HTML= option provides a way to associate Uniform Resource Locators (URLs) with subgroups plotted on a control chart. It specifies a variable in the input data set whose values provide the URLs to be associated with different subgroups. The HTML= variable can be a character variable or a numeric variable with an associated character format.

The following statements generate an $\bar{X}$ chart that is saved to a GIF file and included in an HTML file. The formatted values of the numeric HTML= variable WEB are URLs that link subgroups in the input data set to various web pages.

```
goptions target = gif;
ods html body = "example1.html";

proc format;
   value webfmt
   1='href="http://www.sas.com/"'
   2='href="http://www.sas.com/service/techsup/faq/qc/shewproc.html"'
   3='href="http://www.sas.com/rnd/app/qc.html"'
   4='href="http://www.sas.com/rnd/app/qc/qcnew.html"'
   5='href="http://www.sas.com/rnd/app/qc/qc.html"'
   ;

data wafers;
   format web webfmt.;
   input batch web @;
   do i=1 to 5;
      input diamtr @;
      output;
      end;
   drop i;
datalines;
 1  1  35.00 34.99 34.99 34.98 35.00
 2  1  35.00 34.99 34.99 34.98 35.00
 3  1  34.99 34.99 35.00 34.99 35.00
 4  1  35.00 35.00 34.99 34.99 35.00
 5  2  35.00 34.99 34.98 34.99 35.00
 6  2  34.99 34.99 35.00 35.00 35.00
 7  2  35.01 34.98 35.00 35.00 34.99
 8  2  35.00 35.00 34.99 34.98 34.99
 9  3  34.99 34.98 34.99 35.01 35.00
10  3  34.99 35.00 35.00 34.99 35.00
11  3  35.01 35.00 35.00 34.98 34.99
12  3  34.99 34.99 35.00 34.98 35.01
13  4  35.01 34.99 34.98 34.99 34.99
14  4  35.00 35.00 34.99 35.00 34.99
15  4  34.98 35.00 34.99 35.00 34.99
16  4  34.99 35.00 35.00 35.01 35.00
17  5  34.98 34.98 34.98 34.99 34.98
18  5  35.01 35.02 35.00 34.98 35.00
19  5  34.99 34.98 35.00 34.99 34.98
20  5  34.99 35.00 35.00 34.99 34.99
;

symbol1 v=square;
proc shewhart data=wafers;
   xchart diamtr*batch / html = ( web );
run;

ods html close;
run;
```

In this example five different URLs are each associated with a set of four subgroup values. When you view the ODS HTML output with a browser, you can click on

a subgroup data point and the browser will bring up the page specified by the subgroup's URL. These URLs happen to point to pages at SAS Institute's web site which may be of interest to SAS/QC users.

Note: The value of the HTML= variable must be the same for each observation belonging to a given subgroup.

URLS and Tests for Special Causes

The TESTURLS= data set provides a way to associate a URL with each subgroup in a control chart for which a given test for special causes is positive:

Table 57.2. Variables Required in a TESTURLS= Data Set

Variable	Type	Description
TEST	character or numeric	test identifier
CHART	numeric	primary (1) or secondary (2) chart
URL	character	URL associated with subgroups with positive test

The variable _TEST_ identifies a test for special causes (see Chapter 55, "Tests for Special Causes," on page 1975). A standard test is identified by its number (1 to 8) and a nonstandard test is identified by the CODE= character in its pattern specification. The _TEST_ variable must be a character variable if nonstandard tests are included in the TESTURLS= data set. The value of _CHART_ is 1 or 2, specifying whether the test applies to the primary or secondary chart. The character variable _URL_ contains the URL link to be associated with subgroups for which the test is positive.

The following statements create a TESTURLS= data set and an $\bar{X}$ chart using the same DATA= data set as the previous example:

```
ods html body = "example2.html";

data testlink;
     length _URL_ $ 75;
     input _TEST_ _CHART_ _URL_;
datalines;
1  1  href="http://www.sas.com/"
2  1  href="http://www.sas.com/service/techsup/faq/qc/shewproc.html"
3  1  href="http://www.sas.com/rnd/app/qc.html"
4  1  href="http://www.sas.com/rnd/app/qc/qcnew.html"
5  1  href="http://www.sas.com/products/qc/index.html"
6  1  href="http://www.sas.com/rnd/app/qc/qc/qcspc.html"
7  1  href="http://www.sas.com/software/components/qc.html"
8  1  href="http://www.sas.com/rnd/app/qc/qc.html"
;
```

```
symbol1 v=dot;
proc shewhart data=wafers testurls=testlink;
   xchart diamtr*batch / tests = 1 to 8;
run;

ods html close;
run;
```

In this example only subgroups triggering tests for special causes have URLs associated with them.

Note: If a TESTSURLS= data set and an HTML= variable are both specified, the URL from the TESTURLS= data set is associated with any subgroup for which the test is positive.

References

Al-Salti, M. and Statham, A. (1994), "A Review of the Literature on the Use of SPC in
Batch Production," *Quality and Reliability Engineering International*, 10, 49–62.

Alt, F. (1985), "Multivariate Quality Control," *Encyclopedia of Statistical Sciences, Volume 6*, edited by S. Kotz and N. L. Johnson, New York: John Wiley & Sons, Inc., 110–122.

Alwan, L. C. and Roberts, H. V. (1988), "Time Series Modeling for Statistical Process Control," *Journal of Business and Economic Statistics*, 6, 87–95.

American Society for Testing and Materials (1976), *ASTM Manual on Presentation of Data and Control Chart Analysis*, 1916 Race Street, Philadelphia, PA 19103.

ASQC Automotive Division/AIAG (1990), *Fundamental Statistical Process Control: Reference Manual,* AIAG.

Austin, J. A. (1973), "Control Chart Constants for Largest and Smallest in Sampling from a Normal Distribution Using the Generalized Burr Distribution," *Technometrics,* 15, 931–933.

Bissell, A. F. (1990), "How Reliable is Your Capability Index?," *Applied Statistics,* 39, No. 3, 331–340.

Boyles, R. A. (1997), "Estimating Common-Cause Sigma in the Presence of Special Causes," *Journal of Quality Technology,* 29, 381–395.

Box, G. E. P. and Jenkins, G. M. (1976), *Time Series Analysis: Forecasting and Control*, San Francisco: Holden-Day.

Box, G. E. P. and Kramer, T. (1992), "Statistical Process Monitoring and Feedback Adjustment—A Discussion," *Technometrics*, 34, 251–285 (with discussion).

Burr, I. W. (1969), "Control Charts for Measurements with Varying Sample Sizes," *Journal of Quality Technology,* 1, 163–167.

Burr, I. W. (1976), *Statistical Quality Control Methods,* New York: Marcel Dekker, Inc.

Champ, S. W. and Woodall, W. H. (1987), "Exact Results for Shewhart Control Charts With Supplementary Runs Rules," *Technometrics*, 29, 393–401.

Champ, S. W. and Woodall, W. H. (1990), "A Program to Evaluate the Run Length Distribution of a Shewhart Control Chart with Supplementary Run Rules," *Journal of Quality Technology*, 29, 393–399.

Deming, W. E. (1982), *Out of the Crisis*, Cambridge, MA: Massachusetts Institute of Technology, Center for Advanced Engineering Study.

Doganaksoy, N., Faltin, F. W., and Tucker, W. T. (1991), "Identification of Out-of-Control Quality Characteristics in a Multivariate Manufacturing Environment," *Communications in Statistics—Theory and Methods*, 20, 2775–2790.

Farnum, N. R. (1992), "Control Charts for Short Runs: Nonconstant Process and Measurement Error," *Journal of Quality Technology*, 24, 138–144.

Gnanadesikan, R. and Kettenring, J. R. (1972), "Robust Estimates, Residuals, and Outlier Detection with Multiresponse Data," *Biometrics,* 28, 81–124.

Grant, E. L. and Leavenworth, R. S. (1988), *Statistical Quality Control, Sixth Edition*, New York: McGraw-Hill.

Hawkins, D. M. (1991), "Multivariate Quality Control Based on Regression-Adjusted Variables," *Technometrics*, 33, 61–75.

Hawkins, D. M. (1993), "Regression Adjustment for Variables in Multivariate Quality Control," *Journal of Quality Technology*, 25, 170–182.

Hillier, F. S. (1969), "$\bar{X}$- and R-Chart Control Limits Based On a Small Number of Subgroups," *Journal of Quality Technology*, 1, 17–26.

Hotelling, H. (1931), "The Generalization of Student's Ratio," *Annals of Mathematical Statistics*, 2, 360–378.

Hotelling, H. (1947), "Multivariate Quality Control," *Techniques of Statistical Analysis* (C. Eisenhart, M. Hastay, and W. A. Wallis, eds.), New York: McGraw-Hill, 111–184.

Hunter, J. S. (1986), "The Exponentially Weighted Moving Average," *Journal of Quality Technology,* 18, 203–210.

Hunter, J. S. (1988), "The Digidot Plot," *The American Statistician*, 24, 54.

Iglewicz, B. and Hoaglin, D. (1987), "Use of Boxplots for Process Evaluation," *Journal of Quality Technology*, 19, 180–190.

Jackson, J. E. (1980), "Principal Components and Factor Analysis: Part I —Principal Components," *Journal of Quality Technology*, 12, 201–213.

Jackson, J. E. (1985), "Multivariate Quality Control," *Communications in Statistics*, 14 (11), 2657–2688.

Jackson, J. E. (1991), *A User's Guide to Principal Components*, New York: John Wiley & Sons, Inc.

Johnson, N. L., Kotz, S., and Kemp, A. W. (1992) *Univariate Discrete Distributions, Second Edition,* New York: John Wiley & Sons, Inc.

Kume, H. (1985), *Statistical Methods for Quality Improvement*, Tokyo: AOTS Chosakai, Ltd.

MacGregor, J. (1987), "Interfaces Between Process Control and Online Statistical Process Control," *Computing and Systems Technology Division Communications*, 10, 9–20.

MacGregor, J. (1990), "A Different View of the Funnel Experiment," *Journal of Quality Technology*, 22, 255–259.

MacGregor, J., Hunter, J. S., and Harris, T. (1988), "SPC Interfaces," short course notes.

McGill, R., Tukey, J. W., and Larsen, W. A. (1978), "Variations of Box Plots," *The American Statistician,* 32, 12–16.

Mason, R. L., Tracy, N. D., and Young, J. C. (1993), "Use of Hotelling's T^2 Statistic in Multivariate Control Charts," unpublished manuscript.

Montgomery, D. C. (1996), *Introduction to Statistical Quality Control, Third Edition*, New York: John Wiley & Sons, Inc.

Montgomery, D. C. and Mastrangelo, C. M. (1991), "Some Statistical Process Control Methods for Autocorrelated Data," *Journal of Quality Technology*, 23, 179–204 (with discussion).

Montgomery, D. C., Keats, J. B., Runger, G. C. and Messina, W. S. (1994), "Integrating Statistical Process Control and Engineering Process Control," *Journal of Quality Technology*, 26, 79–87.

Nelson, L. S. (1982), "Control Charts for Individual Measurements," *Journal of Quality Technology*, 14, 172-174.

Nelson, L. S. (1984), "The Shewhart Control Chart—Tests for Special Causes," *Journal of Quality Technology*, 15, 237–239.

Nelson, L. S. (1985), "Interpreting Shewhart $\bar{X}$ Control Charts," *Journal of Quality Technology*, 17, 114–116.

Nelson, L. S. (1989), "Standardization of Shewhart Control Charts," *Journal of Quality Technology*, 21, 287–289.

Nelson, L. S. (1990), "Setting Up a Control Chart Using Subgroups of Varying Sizes," *Journal of Quality Technology*, 22, 245–246.

Nelson, L. S. (1994), "Shewhart Control Charts With Unequal Subgroup Sizes," *Journal of Quality Technology*, 26, 64–67.

Quesenberry, C. P. (1991a), "SPC Q Charts for Start-Up Processes and Short or Long Runs," *Journal of Quality Technology*, 23, 213–224.

Quesenberry, C. P. (1991b), "SPC Q Charts for a Binomial Parameter p: Short or Long Runs," *Journal of Quality Technology*, 23, 239–246.

Quesenberry, C. P. (1993), "The Effect of Sample Size on Estimated Effects," *Journal of Quality Technology*, 25, 237–247.

Rocke, D. M. (1989), "Robust Control Charts," *Technometrics*, 31, 173–184.

Rodriguez, R. N. and Bynum, R. A. (1992), *Examples of Short Run Process Control Methods With the SHEWHART Procedure in SAS/QC Software*. Unpublished manuscript available from the authors.

Rodriguez, R. N. and Bynum, R. A. (1993), *Process Capability Analysis Using SAS/QC Software, Release 6.08*. Unpublished manuscript available from the authors.

Ryan, T. P. (1989), *Statistical Methods for Quality Improvement*, New York: John Wiley & Sons, Inc.

SAS Institute Inc. (1999), *SAS/GRAPH Software: Reference, Version 8*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1989b), SAS Technical Report P-188: *SAS/QC Software Examples, Version 6*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1999), *SAS Language Reference: Dictionary, Version 8*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1991a), SAS Technical Report P-229: *SAS/STAT Software: Changes and Enhancements, Release 6.07,* Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1991b), *SAS/QC Software: SQC Menu System, Version 6, First Edition*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1999), *SAS/ETS User's Guide, Version 8*, Cary, NC: SAS Institute Inc.

Schilling, E. G. and Nelson, P. R. (1976), "The Effect of Non-Normality on the Control Limits of $\bar{X}$ Charts," *Journal of Quality Technology*, 8, 183–187.

Schneider, H. and Pruett, J. M. (1994), "Control Charting Issues in the Process Industries," *Quality Engineering*, 6, 347–373.

Shewhart, W. A. (1931), *Economic Control of Quality Manufactured Product*, New York: D. Van Nostrand Company, Inc. Republished in 1980 by the American Society of Quality Control.

Snedecor, G. W. and Cochran, W. G. (1980), *Statistical Methods, Seventh Edition*, Ames, IA: The Iowa State University Press.

Teichroew, D. (1956), "Tables of Expected Values of Order Statistics and Products of Order Statistics for Samples of Size 20 and Less from the Normal Distribution," *Annals of Mathematical Statistics,* 27, 410–426. Reproduced in Sarhan, A. E. and Greenberg, B. G. (eds.) (1962), *Contributions to Order Statistics,* New York: John Wiley & Sons, Inc.

Tracy, N. D., Young, J. C., and Mason, R. L. (1992), "Multivariate Control Charts for Individual Observations," *Journal of Quality Technology*, 24, 88–95.

Tukey, J. W. (1977), *Exploratory Data Analysis*, Reading, MA: Addison-Wesley.

Western Electric Company (1956), *Statistical Quality Control Handbook*, available from Western Electric Company, Commercial Sales Clerk, Select Code 700-444, P. O. Box 26205, Indianapolis, Indiana 46226.

Wetherill, G. B. and Brown, D. W. (1991), *Statistical Process Control: Theory and Practice*, London: Chapman and Hall.

Wheeler, D. J. (1991a), *Short Run SPC*, Knoxville, TN: SPC Press, Inc.

Wheeler, D. J. (1991b), "Shewhart's Chart: Myths, Facts, and Competitors," *45th Annual Quality Congress Transactions*, American Society for Quality Control. 533–538.

Wheeler, D. J. (1995), *Advanced Topics in Statistical Process Control*, Knoxville, TN: SPC Press, Inc.

Wheeler, D. J. and Chambers, D. S. (1986), *Understanding Statistical Process Control*, Knoxville, TN: SPC Press, Inc.

Wilks, S. S. (1962), *Mathematical Statistics,* New York: John Wiley & Sons, Inc.

Woodall, W. H. (1993), "Autocorrelated Data and SPC," *ASQC Statistics Division Newsletter*, 13, 18–21.

Part 11
Appendices

Contents

Appendix A. The GAGE Application 2057

Appendix B. The RELIABILITY Graphical Interface 2085

Appendix C. Functions 2089

Appendix D. Special Fonts in SAS/QC Software 2117

Appendix E. References 2123

Appendix A
The GAGE Application

Appendix Contents

INTRODUCTION 2059
Terminology 2059

GETTING STARTED 2060
Invoking the GAGE Application 2061
Entering Data 2062
Performing a Range Chart Analysis 2064
Performing an Average Chart Analysis 2066
Selecting a Statistical Method 2067
Performing an Average and Range Analysis 2067
Performing a Variance Components Analysis 2069
Saving the Data 2070
Entering Another Set of Data 2071
Reading Data from a Data Set 2071

DETAILS 2072
Range Chart 2072
Average Chart 2073
Average and Range Method 2074
Variance Components Method 2078
Creating a Data Set Outside the GAGE Application 2080
Extensibility of the Application 2083

Appendix A
The GAGE Application

This appendix describes the GAGE application, which is a tool for assessing gage repeatability and reproducibility (R&R). The GAGE application is available with Release 6.10 of the SAS/QC Sample Library.

Introduction

Measurement systems are essential to the quality of a manufacturing process. The gages or instruments that take measurements are subject to variation. Too much variation in the measurement system may mask variation in the process.

One type of measurement variation is caused by conditions inherent in gages. This variation, known as repeatability, is obtained when one person measures the same characteristic several times with the same gage. Another type of measurement variation, known as reproducibility, occurs when different individuals measure the same characteristic with the same gage. Other sources of measurement variation include part-to-part variation, accuracy, stability, and linearity.

Two graphical methods for evaluating the measurement system are range charts and average charts. Range charts assess repeatability by showing whether the gage variability is consistent. Average charts show consistency of operator variability (reproducibility) and part-to-part variation.

Two statistical approaches to determining gage R&R are the average and range method and the variance components method. The variance components method can provide more information, is more accurate, and is more flexible than the average and range method.

The GAGE application makes it easy for you to enter your data, create range and average charts, and determine gage R&R. Whether you use the average and range method or the more powerful variance components method, the GAGE application reports the results in a standard form. It allows you to save the graphs for later reference, and it allows you to save the reports. You can save the data in a SAS data set for subsequent gage analysis or for more extensive analysis using other components of the SAS System. Because gage R&R techniques are open to local interpretation, this application has been designed so that it can be modified to suit the needs of your company.

Terminology

The following definitions describe the terms used in a gage R&R study.

Gage — any device used to obtain measurements, for example, a micrometer or a gasket thickness gage.

Condition	typically an operator, but can be thought of more generically as any condition affecting the measurements. For example, with an automated process, condition might be a set-up procedure or an environmental condition such as temperature. For the remainder of this appendix, condition is referred to as operator.
Trial	a set of measurements on all parts taken by one operator. Multiple trials help separate the gage variability (repeatability) from the variability contributed by operators (reproducibility).
Part	the item that is measured, for example, a gasket. The parts selected should represent the entire operating range (variability) of the process.
Measurement System	the complete process used to obtain measurements. This includes people, gages, operations, and procedures.
Repeatability	the variation resulting from repeated measurements taken on the same part with the same gage by the same operator. Repeatability is the gage or equipment variation.
Reproducibility	the variation in the average of the measurements resulting when different operators using the same gage take measurements on the same part. Reproducibility is the operator-to-operator variability.

Getting Started

Suppose that ABC Company needs to evaluate a gasket thickness gage. Three operators (George, Jane, and Robert) are selected for this study. Using the same gage, each operator measures ten parts (gaskets) in a random order. Each part is measured by each operator twice (two trials). Table A.1 gives the measurements (gasket thicknesses) collected by each operator and is patterned after an example given in *Measurement Systems Analysis Reference Manual* (1990).

Table A.1. Gage Study Data

	George		Jane		Robert	
Part	Trial1	Trial2	Trial1	Trial2	Trial1	Trial2
1	0.65	0.60	0.55	0.55	0.50	0.55
2	1.00	1.00	1.05	0.95	1.05	1.00
3	0.85	0.80	0.80	0.75	0.80	0.80
4	0.85	0.95	0.80	0.75	0.80	0.80
5	0.55	0.45	0.40	0.40	0.45	0.50
6	1.00	1.00	1.00	1.05	1.00	1.05
7	0.95	0.95	0.95	0.90	0.95	0.95
8	0.85	0.80	0.75	0.70	0.80	0.80
9	1.00	1.00	1.00	0.95	1.05	1.05
10	0.60	0.70	0.55	0.50	0.85	0.80

These data are used to illustrate the GAGE application throughout this appendix.

Invoking the GAGE Application

The interface to the GAGE application was implemented using FRAME entries in SAS/AF software. The application is stored in the gage catalog. (File extensions for SAS catalogs differ based on the operating system.)

Assume that you are using the SAS System under Microsoft Windows and that the SAS/QC Sample Library is stored in the c:\sas\qc\sample directory. (Check with your SAS site representative for the location of the Sample Library on your system.) You invoke the application as follows:

1. First you must tell the SAS System where the catalog is stored:
   ```
   libname gage 'c:\sas\qc\sample';
   ```
2. You then issue the following command from any SAS display manager window:
   ```
   af c=gage.gage.gage.frame
   ```

The main window in the application appears, as shown in Figure A.1.

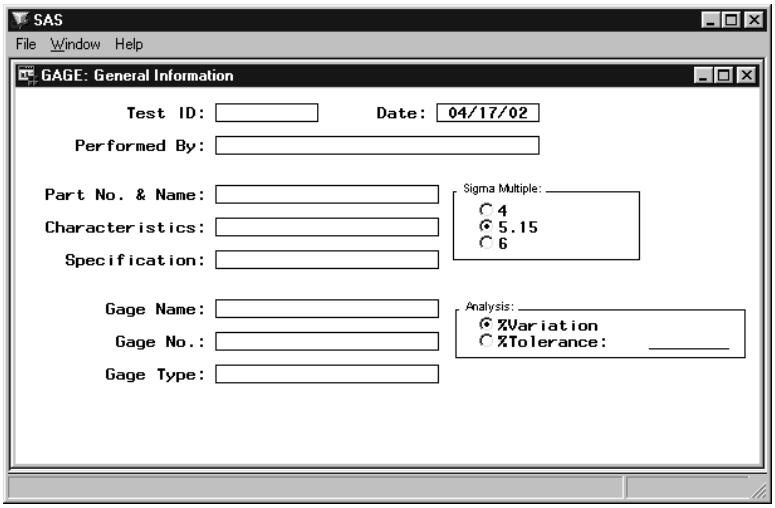


Figure A.1. General Information Window

You specify the following general information for your gage study in this window: who performed the study, part number and part name, characteristics, specification, and gage name, number, and type.

The test ID is required for each set of data to be analyzed. It uniquely identifies each gage study. Date is set to the current date and can be changed. You also select either the percent of process variation analysis (%Variation) or the percent of tolerance analysis (%Tolerance). You must specify a tolerance value for the percent of tolerance analysis.

Set the multiple of sigma to whatever level you use for gage studies. This multiple may be a standard level established within your organization. For example, the automotive industry typically uses 5.15σ (ASQC Automotive Division/AIAG 1990), and SEMATECH uses 6σ (SEMATECH, Inc. 1991).

The general information for the gasket thickness gage example is entered, as shown in Figure A.2.

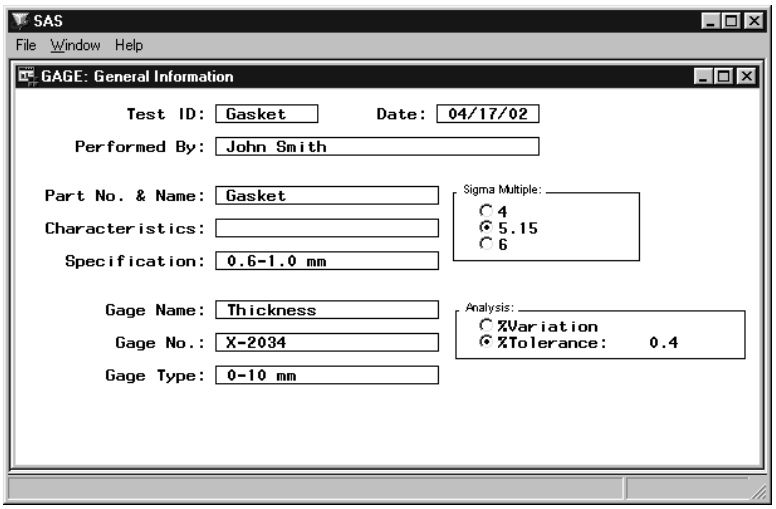


Figure A.2. General Information for Gage Study

This study is identified with the test ID Gasket. The analysis is based on 5.15σ, and a percent of tolerance analysis based on a tolerance of 0.4mm is requested.

Entering Data

The next step is to enter the data provided in Table A.1. In the General Information window, choose Edit from the File menu, as shown in Figure A.3.

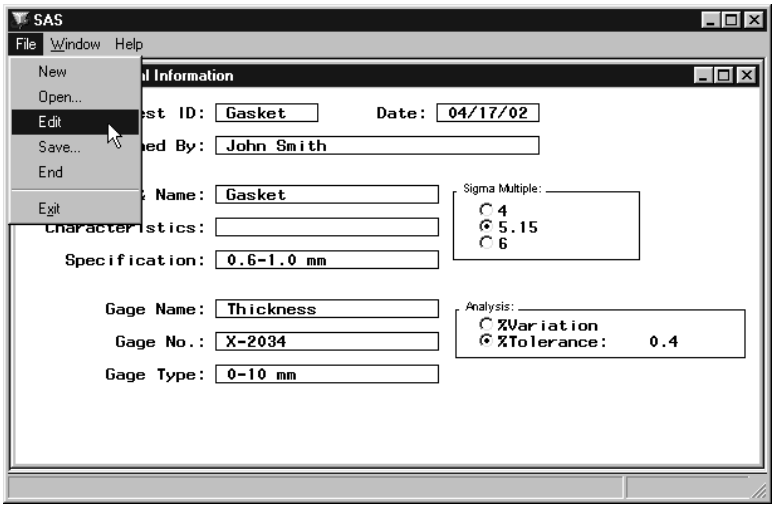


Figure A.3. Choosing Edit from the File Menu

The Measurements window appears, as shown in Figure A.4.

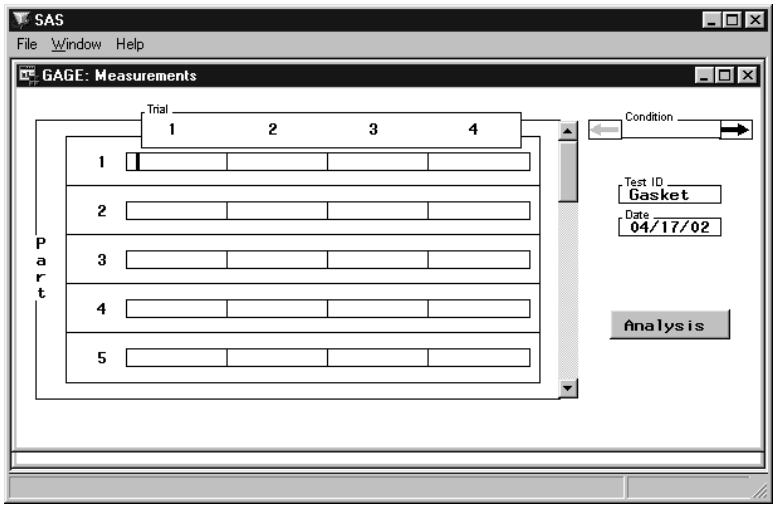


Figure A.4. Measurements (Data Entry) Window

Enter data for one operator (Condition) in this window only. Only five parts are displayed at one time. Use the scroll bar to move the data region vertically. The first operator is George, whose measurements for parts1-5 are shown in Figure A.5.

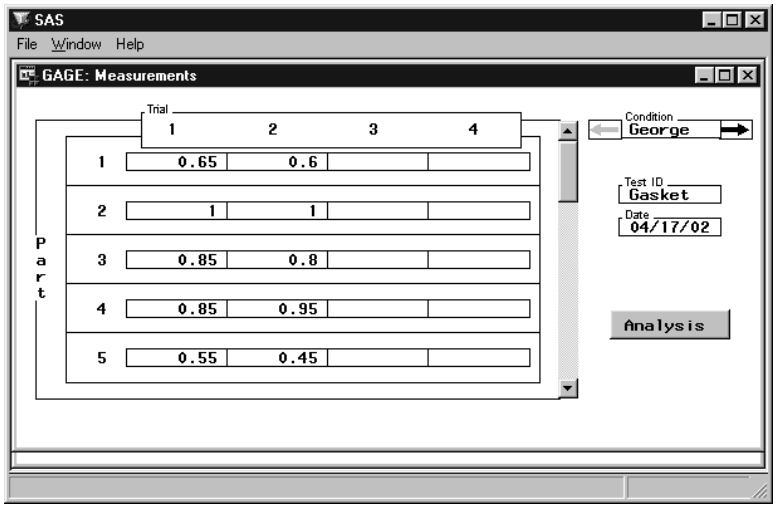


Figure A.5. Measurements for Operator George

To enter the next operator's measurements, press the arrow ⇒ to the right of Condition. An empty Measurements window similar to Figure A.4 appears.

The second operator is Jane. Her measurements for parts 6–10 are shown in Figure A.6.

Data for the third operator are entered similarly. Press the arrow ⇐ to the left of Condition to move to the previous operator.

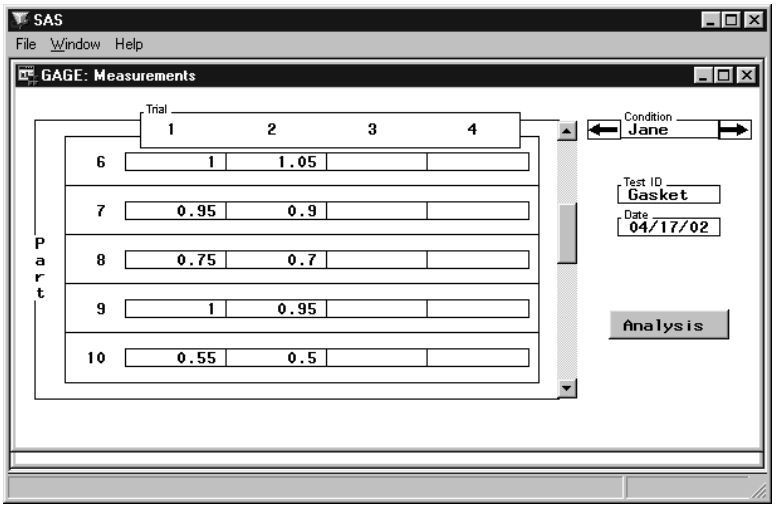


Figure A.6. Measurements for Operator Jane

Performing a Range Chart Analysis

Now that the data are entered, the gage variability (repeatability) can be checked for consistency. This is done graphically with a range chart. Press the Analysis button in the Measurements window. A menu of analysis options appears, as shown in Figure A.7.

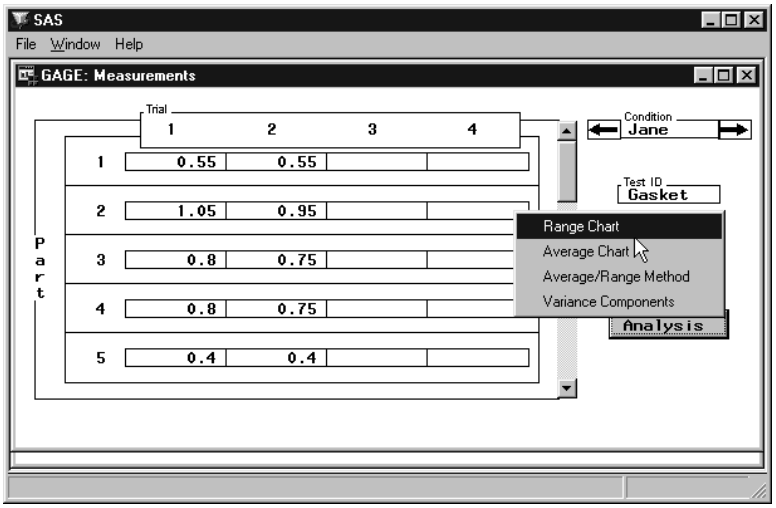


Figure A.7. Analysis Button Options

Select Range Chart. The range chart of the data is displayed, as shown in Figure A.8.

No points are out-of-control, and the variability across operators is fairly comparable. This indicates that all operators are using the gage in the same way. If there were any out-of-control points, they should be investigated and dealt with before proceeding.

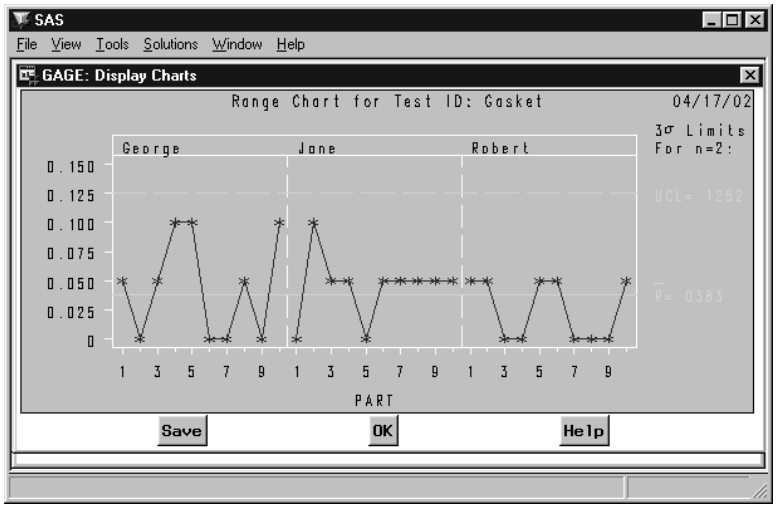


Figure A.8. Range Chart for Gage Study Gasket

You can save this chart in a SAS graphics catalog. Then you can use the GRAPH window or the GREPLAY procedure to view charts stored in the catalog. You also can create hard-copy versions of charts stored in the catalog.

Press the Save button to save the range chart. The Save Charts window appears, as shown in Figure A.9.

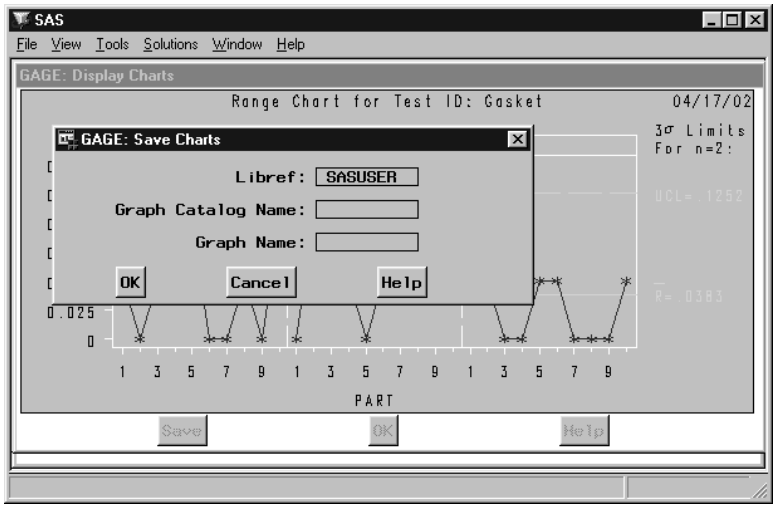


Figure A.9. Saving the Range Chart

Libref — tells the SAS System where to store the SAS graphics catalog. The libref SASUSER stores the catalog in the SASUSER data library, which is created automatically by the SAS System. You must assign other libref locations with the LIBNAME statement.

Graph Catalog Name — is the name of the SAS graphics catalog in which to store the chart.

Graph Name — is the name of this range chart when stored in the SAS graphics catalog.

Figure A.10 shows that the range chart is to be stored in SASUSER.GASKET.RANGE.

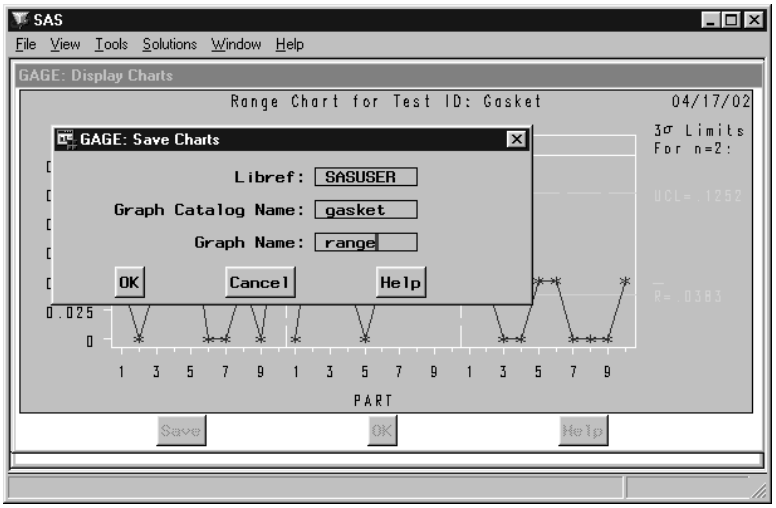


Figure A.10. Saving the Range Chart

Press the Cancel button if you decide not to save the chart. Press the OK button to save the chart.

You return to the window displaying the range chart. Press the OK button to leave the Display Charts window.

Refer to SAS/GRAPH documentation for more information on SAS graphics catalogs, the GRAPH window, and the GREPLAY procedure.

Performing an Average Chart Analysis

The average chart shows operator variability (reproducibility) and part-to-part variation. Press the Analysis button in the Measurements window and select Average Chart. The average chart of the data is displayed, as shown in Figure A.11.

Note how most of the averages are beyond the control limits. Also, the out-of-control averages tend to be the same for each operator. This indicates that this study can detect part-to-part variation. If the averages were not outside the control limits, the part-to-part variation would be hidden in the gage variation.

Operator variability can be seen on the average chart by comparing the operator averages for each part. These averages will differ when there is variability.

Note that this is not a standard use of the Shewhart chart. Ordinarily the fact that the points fall outside the control limits would raise concerns that the process is out of control, but here the opposite conclusion is drawn.

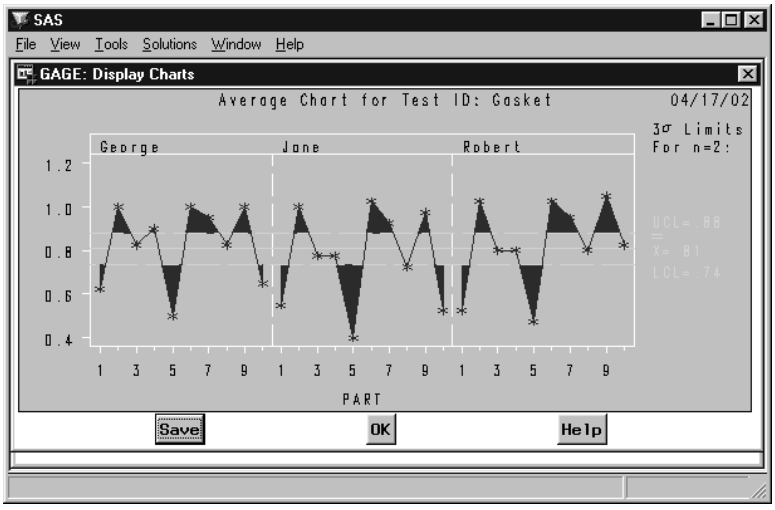


Figure A.11. Average Chart for Gage Study Gasket

You can save the average chart by pressing the Save button, as described for the range chart. Press the OK button to leave this window.

Selecting a Statistical Method

The range chart indicates that the gage variability is consistent. The average chart indicates that the measurement system is adequate to detect part-to-part variation. Now you can perform a statistical analysis. Which analysis method should you choose: average and range, or variance components?

Since the data for this study are balanced (no data points are missing), the average and range method can be used. But it is not as efficient as the variance components method, and it does not provide information about the interaction between operators and parts. Both methods will be shown to illustrate some of these differences.

Performing an Average and Range Analysis

Press the Analysis button in the Measurements window and select Average/Range Method. The results are displayed, as shown in Figure A.12. You can scroll the report with the vertical scroll bar.

The complete listing of these results is shown in Figure A.24 on page 2077. A percent of tolerance analysis (against a tolerance value of 0.4mm) was requested in the General Information window. This appears on the right side of the report.

As with the charts, you can save this report in a file. Press the Save button to save the report. The Save Reports window appears, as shown in Figure A.13.

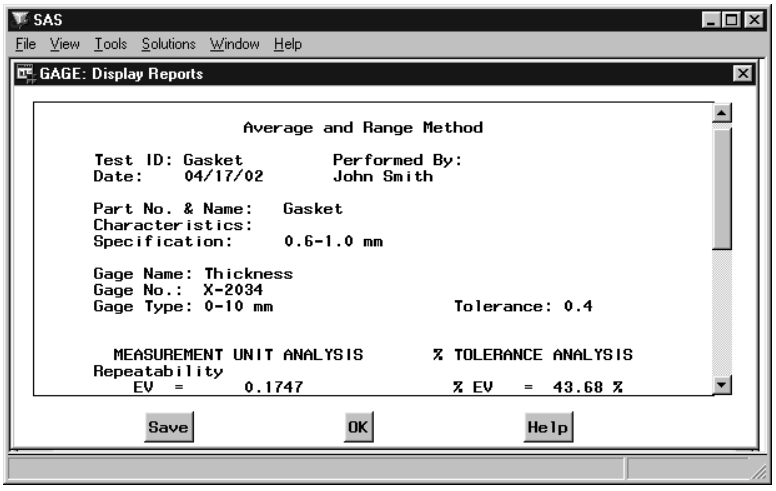


Figure A.12. Average and Range Analysis of Gage Study Gasket

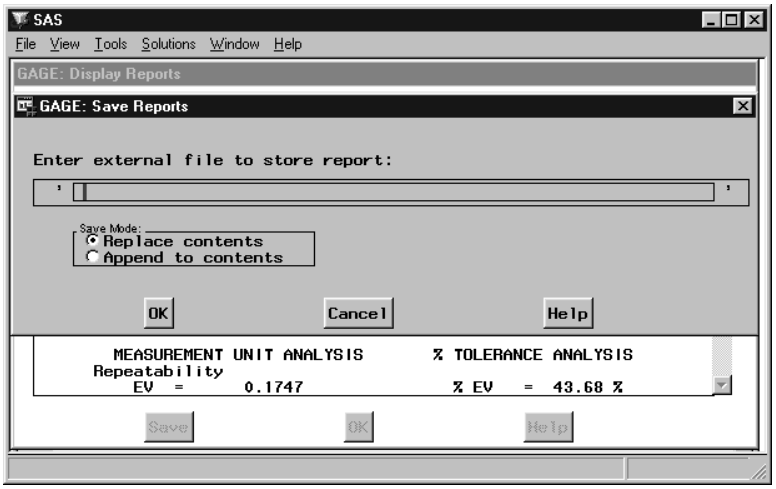


Figure A.13. Saving the Average and Range Report

You enter the name of the file in which to store the report. You can select to have this listing replace what is currently in the file (if it exists) or be appended to the information stored in the file.

Press the Cancel button if you decide not to save the report. Press the OK button to save the report.

You return to the window displaying the average and range report. Press the OK button to leave the Display Reports window.

What would happen if you chose this method and there were missing data? Assume that operator Robert was unable to take the second measurement on the second part, and that data point is missing. If you run the average and range method on these data, you receive the message window shown in Figure A.14.

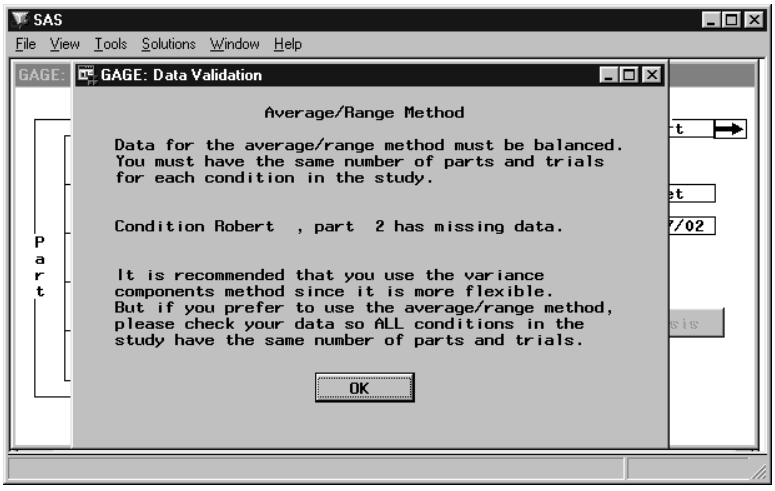


Figure A.14. Average and Range Message Window

The variance components method can be used for these data.

Performing a Variance Components Analysis

Now perform a variance components analysis on the original data. Press the Analysis button in the Measurements window and select Variance Components. The results are displayed, as shown in Figure A.15.

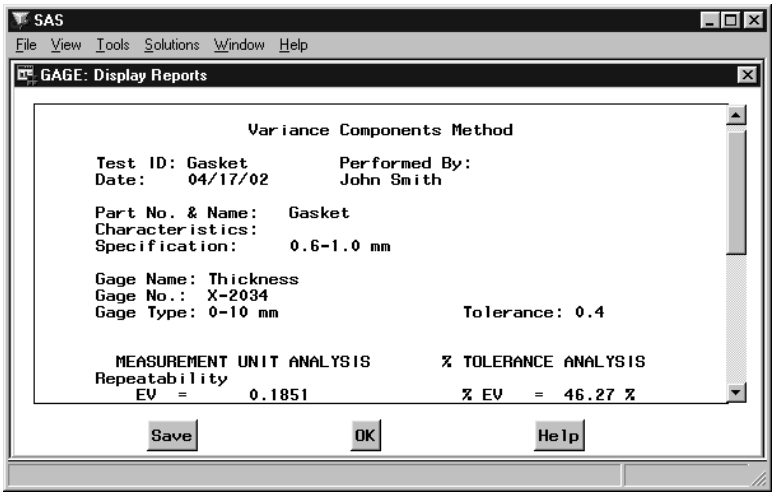


Figure A.15. Variance Components Analysis of Gage Study Gasket

The complete listing of these results is shown in Figure A.25 on page 2078. Note that the results you get using the variance components method differ slightly from those you get using the average and range method (see "Variance Components Method" on page 2078).

You can save this report by pressing the Save button, as described for the average and range analysis. Press the OK button to leave this window.

Saving the Data

Now that you have entered the data and performed analyses, you can save the data in a SAS data set for later analysis. You save the data by choosing Save from the File menu in either the Measurements window or the General Information window.

Choose End from the File menu in the Measurements window to return to the General Information window. Then choose Save from the File menu, as shown in Figure A.16.

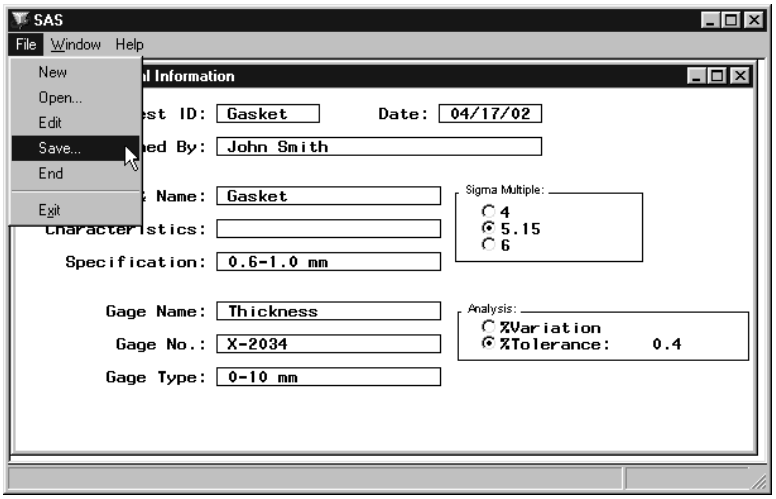


Figure A.16. Choosing Save from the File Menu

The Save Data window appears, as shown in Figure A.17.

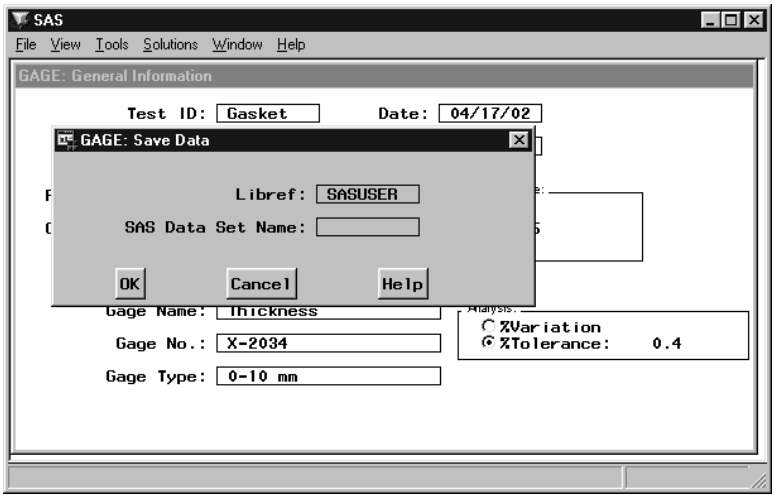


Figure A.17. Saving the Data in a SAS Data Set

Libref — tells the SAS System where to store the SAS data set. The libref SASUSER stores the data set in the SASUSER data library, which is created automatically by the SAS System. You must assign other libref locations with the LIBNAME statement.

SAS Data Set Name is the name of the SAS data set in which to store the data.

Figure A.18 shows that the data are to be stored in SASUSER.GASKET.

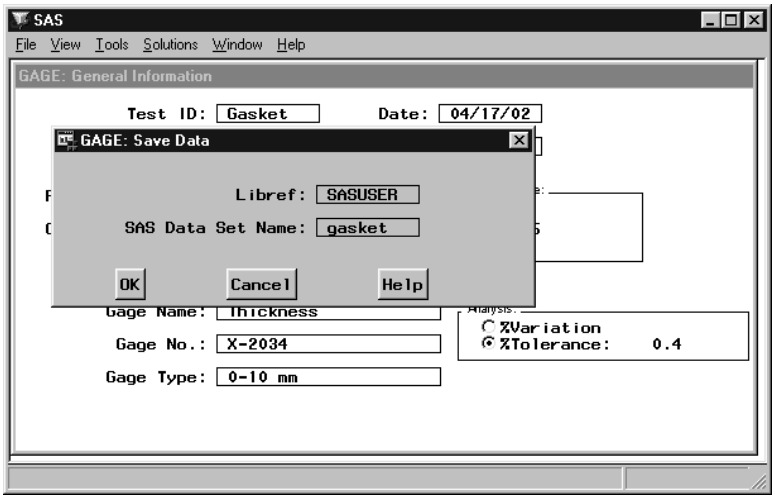


Figure A.18. Saving the Data in a SAS Data Set

Press the Cancel button if you decide not to save the data. Press the OK button to save the data.

You return to the General Information window.

Entering Another Set of Data

Suppose you have more than one gage study to analyze. You need to clear the current data before entering the new information. You do so by choosing New from the File menu in the General Information window.

Reading Data from a Data Set

You can save your gage study data in a SAS data set, but how do you bring it back in for further analysis?

Choose Open from the File menu in the General Information window to read data into the GAGE application. A directory of available SAS data sets appears, as shown in Figure A.19.

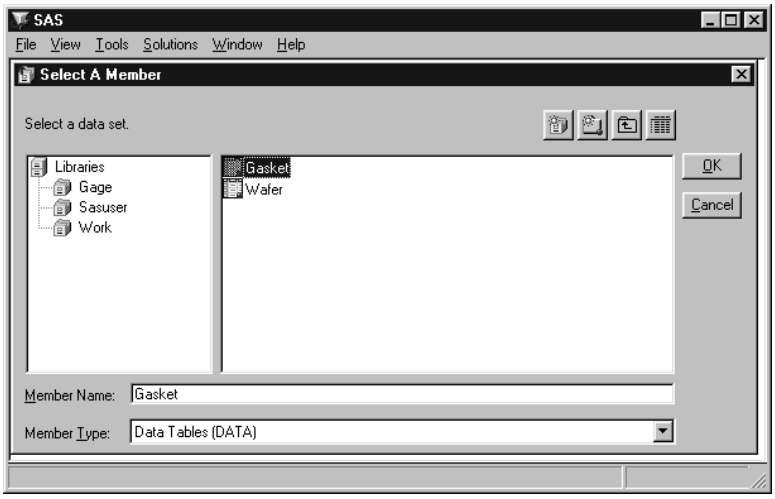


Figure A.19. Selecting a SAS Data Set

Select the data set name from the list. The general information associated with the data is displayed, as shown in Figure A.20.

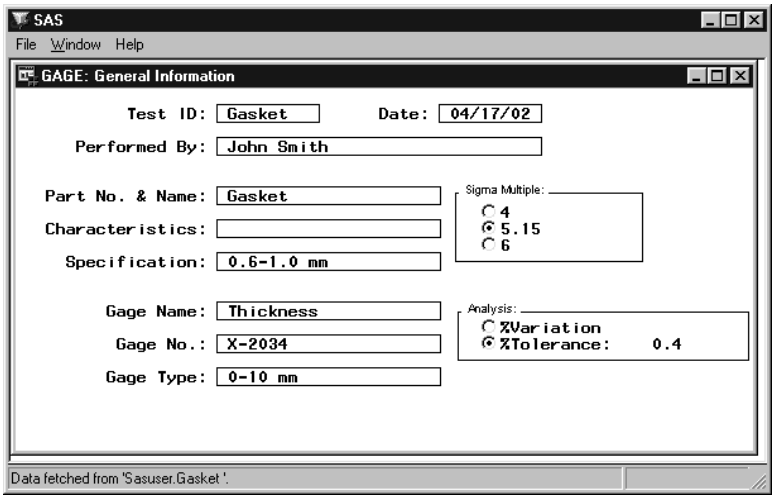


Figure A.20. Data Read from a SAS Data Set

Note: The GAGE application reads only SAS data sets that you have previously created using the application.

Details

Range Chart

The range chart is a graphical method for assessing repeatability. It indicates whether the gage variability is consistent. The ranges for each part and each operator are displayed, as shown in Figure A.21.

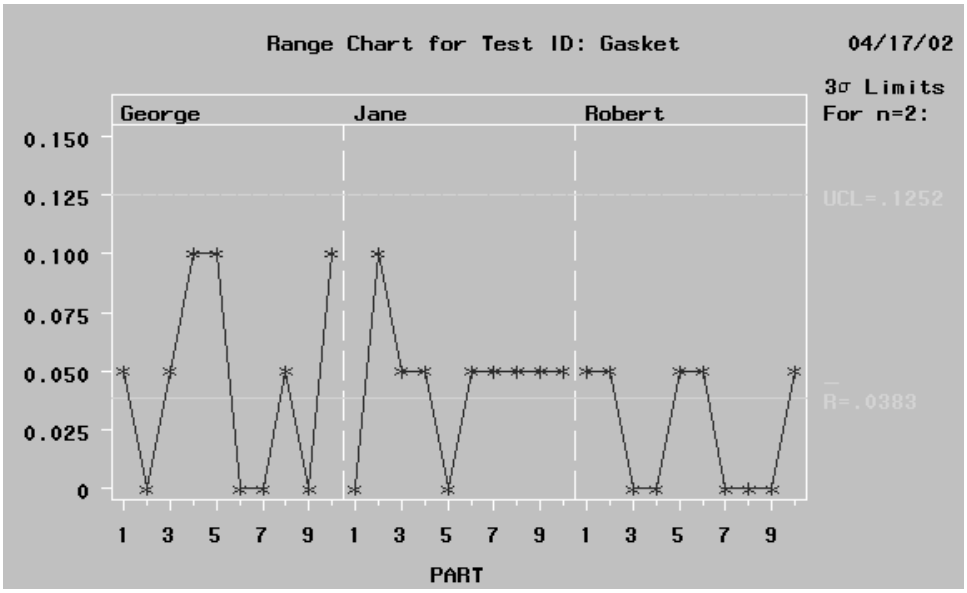


Figure A.21. Range Chart

For example, in Table A.1 on page 2060 the range for operator George, part 1 is calculated as $0.65 - 0.60 = 0.05$. Similarly computed ranges are displayed for each operator and each part.

You should investigate out-of-control points and deal with them before proceeding. If you notice that one operator's ranges are out of control, it implies that his technique differs from the others. If all operators have some out-of-control ranges, you might conclude that operator technique is affecting the measurement system and investigate the need for training.

The range chart is created with the RCHART statement of the SHEWHART procedure. For further information, see Chapter 46, "RCHART Statement."

Average Chart

The average control chart shows both the consistency of operator variability (reproducibility) and part-to-part variation. The averages of the measurements for each part and each operator are displayed, as shown in Figure A.22.

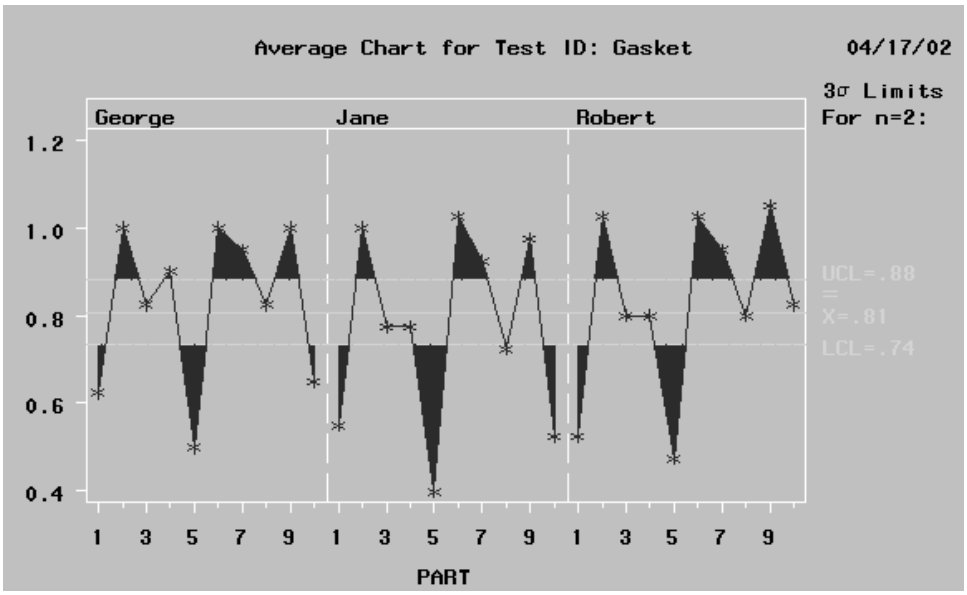


Figure A.22. Average Chart

For example, in Table A.1 on page 2060 the average for operator Jane, part 4 is calculated as $(0.80 + 0.75)/2 = 0.775$. Similarly computed averages are displayed for each operator and each part.

Operator variability can be seen by comparing the operator averages for each part. These averages will differ when there is variation.

The average chart also shows part-to-part differences. The averages should fall outside the control limits since the control limits are based on gage error (repeatability). If the averages do not fall outside the control limits, the part-to-part variation is hidden in the gage variation. The average chart shows the ability of the measurement system to measure parts. The measurement system is generally considered adequate if most of the averages fall outside the limits and if the out-of-control averages tend to be the same for each operator.

Note that this is not a standard use of the Shewhart chart. Ordinarily the fact that the points fall outside the control limits would raise concerns that the process is out of control, but here the opposite conclusion is drawn.

The average chart is created with the XCHART statement of the SHEWHART procedure. For further information, see Chapter 49, "XCHART Statement."

Average and Range Method

The average and range method is widely used in industry because its calculations can be done by hand. It measures both repeatability and reproducibility for a measurement system.

All calculations described here are based upon a specified multiple of σ, where the multiple ν can be 4, 5.15, or 6. Figure A.23 shows a sample gage report created with

the GAGE application using the average and range method.

The measure of repeatability (or equipment variation), denoted by EV, is calculated as

$$EV = \overline{R} \times K_1$$

where $\overline{R}$ is the average range and K_1 is the adjustment factor

$$K_1 = \frac{\nu}{d_2}$$

```
                Average and Range Method

Test ID: Gasket          Performed By:
Date:    04/17/02        John Smith

Part No. & Name:   Gasket
Characteristics:
Specification:     0.6-1.0 mm

Gage Name: Thickness
Gage No.:  X-2034
Gage Type: 0-10 mm


  MEASUREMENT UNIT ANALYSIS       % PROCESS VARIATION
Repeatability
    EV  =      0.1747               % EV   =  18.70 %
Reproducibility
    AV  =      0.1570               % AV   =  16.80 %
Gage R&R
    R&R =      0.2349               % R&R  =  25.14 %
Part Variation
    PV  =      0.9042               % PV   =  96.79 %
Total Variation
    TV  =      0.9342


     Results are based upon predicting 5.15 sigma.
(99.0% of the area under the normal distribution curve)
```

Figure A.23. Average and Range Method Sample Report

The quantity d_2 (Duncan 1974, Table M) depends on the number of trials used to calculate a single range. In the GAGE application, the number of trials can vary from 2 to 4. Use of d_2 is valid when $\#operators \times \#parts \geq 16$; otherwise, the GAGE application uses d_2^* (Duncan 1974, Table D3), which is based on the number of ranges calculated from $\#operators \times \#parts$ and on the number of trials.

The measure of reproducibility (or appraiser variation), denoted by AV, is calculated as

$$AV = \sqrt{(\overline{X}_{diff} \times K_2)^2 - \frac{(EV)^2}{nr}}$$

where $\overline{X}_{diff}$ is the difference between the maximum operator average and the minimum operator average, K_2 is the adjustment factor

$$K_2 = \frac{\nu}{d_2^*}$$

n is the number of parts, and r is the number of trials. Reproducibility is contaminated by gage error and is adjusted by subtracting $(EV)^2/nr$. The quantity d_2^* (Duncan 1974, Table D3) depends on the number of operators used to calculate a single range. In the GAGE application, the number of operators can vary from 1 to 4. When there is only one operator, reproducibility is set to zero.

The measure of repeatability and reproducibility, denoted by $R\&R$, is calculated as

$$R\&R = \sqrt{(EV)^2 + (AV)^2}$$

Part-to-part variation, denoted by PV, is calculated as

$$PV = R_p \times K_3$$

where R_p is the range of part averages and K_3 is the adjustment factor

$$K_3 = \frac{\nu}{d_2^*}$$

Here the quantity d_2^* (Duncan 1974, Table D3) depends on the number of parts used to calculate a single range. In the GAGE application, the number of parts can vary from 2 to 15.

Total variation, denoted by TV, is based on gage R&R and part-to-part variation.

$$TV = \sqrt{(R\&R)^2 + (PV)^2}$$

The measures of repeatability, reproducibility, gage R&R, part variation, and total variation are shown in Figure A.23 under the heading "MEASUREMENT UNIT ANALYSIS." The right-hand side of the report shows the "% PROCESS VARIATION" analysis, which compares the gage factors to total variation. The percent of total variation accounted for by each factor is calculated as follows:

$$\begin{aligned}
\%EV &= 100\left[\frac{EV}{TV}\right] \\
\%AV &= 100\left[\frac{AV}{TV}\right] \\
\%R\&R &= 100\left[\frac{R\&R}{TV}\right] \\
\%PV &= 100\left[\frac{PV}{TV}\right]
\end{aligned}$$

Note that the sum of these percentages does not equal 100%. You can use these percentages to determine whether the measurement system is acceptable for its intended application.

Instead of percent of process variation, your analysis may be based on percent of tolerance. For this you must specify a tolerance value. Then $\%EV$, $\%AV$, $\%R\&R$, and $\%PV$ are calculated by substituting the tolerance value for TV (the denominator) in the preceding formulas. A sample report with "% TOLERANCE ANALYSIS" is shown in Figure A.24.

What is considered acceptable for $\%R\&R$? Barrentine (1991) gives the following guidelines:

10% or less	excellent
11% to 20%	adequate
21% to 30%	marginally acceptable
over 30%	unacceptable

In general, interpretation may be guided by local standards.

```
                Average and Range Method

Test ID: Gasket          Performed By:
Date:    04/17/02        John Smith

Part No. & Name:   Gasket
Characteristics:
Specification:     0.6-1.0 mm

Gage Name: Thickness
Gage No.:  X-2034
Gage Type: 0-10 mm                    Tolerance: 0.4


  MEASUREMENT UNIT ANALYSIS       % TOLERANCE ANALYSIS
Repeatability
    EV  =      0.1747               % EV   =  43.68 %
Reproducibility
    AV  =      0.1570               % AV   =  39.25 %
Gage R&R
    R&R =      0.2349               % R&R  =  58.72 %
Part Variation
    PV  =      0.9042               % PV   = 226.05 %
Total Variation
    TV  =      0.9342


     Results are based upon predicting 5.15 sigma.
(99.0% of the area under the normal distribution curve)
```

Figure A.24. Average and Range Method with % Tolerance Analysis

Variance Components Method

As an alternative to the average and range method, you can use the variance components method, which is a more powerful statistical technique for analyzing measurement error and other sources of variation in a gage study. Until recently, this method was underutilized for gage studies because it is computationally more difficult to carry out than the average and range method.

Moreover, the language of variance components analysis is alien to most engineers. To overcome this obstacle, the GAGE report for the variance components method is displayed in the same format as that of the average and range method. This format is a modification of the gage repeatability and reproducibility report given in *Measurement Systems Analysis Reference Manual* (1990). Figure A.25 is a sample GAGE report using the variance components method.

As with the average and range method, calculations for the variance components method are based upon a specified multiple of σ, where the multiple ν can be 4, 5.15, or 6.

The advantages of this method versus the average and range method are:

- Variance components are estimated more efficiently in a statistical sense.
- More information can be obtained, such as the interaction between operators and parts (see Figure A.25).
- There are fewer restrictions on the data.

```
                Variance Components Method

Test ID: Gasket          Performed By:
Date:    04/17/02        John Smith

Part No. & Name:   Gasket
Characteristics:
Specification:     0.6-1.0 mm

Gage Name: Thickness
Gage No.:  X-2034
Gage Type: 0-10 mm                    Tolerance: 0.4


  MEASUREMENT UNIT ANALYSIS       % TOLERANCE ANALYSIS
Repeatability
    EV  =      0.1851               % EV   =  46.27 %
Reproducibility
    AV  =      0.1555               % AV   =  38.88 %
Part x Condition
    IV  =      0.2434               % IV   =  60.85 %
Gage R&R
    R&R =      0.3431               % R&R  =  85.77 %
Part Variation
    PV  =      0.9928               % PV   = 248.20 %
Total Variation
    TV  =      1.0504


     Results are based upon predicting 5.15 sigma.
(99.0% of the area under the normal distribution curve)
```

Figure A.25. Variance Components Method Sample Report

The variance components method in the GAGE application uses the MIXED procedure in SAS/STAT software. The MIXED procedure fits mixed linear models, which are a generalization of the standard linear model used in the GLM procedure. Refer to *SAS/STAT User's Guide* for further information on PROC MIXED.

When there is only one operator, PART is a random effect. The MIXED procedure estimates the variance component for PART and for the residual variance (equipment variation) using restricted maximum likelihood (REML).

```
proc mixed;
   class part;
   model meas=;
   random part;
run;
```

When there is more than one operator, there are three random effects: OPERATOR, PART, and OPERATOR*PART. The MIXED procedure uses REML to estimate variance components for these and for the residual variance (equipment variation).

```
proc mixed;
   class operator part;
   model meas=;
   random operator part operator*part;
run;
```

The MIXED procedure creates a table of covariance parameter estimates, including

- σ^2_{EV} the variance component due to equipment variation
- σ^2_{AV} the variance component due to operator variation
- σ^2_{IV} the variance component due to the interaction of operators and parts
- σ^2_{PV} the variance component due to part variation

From these estimates, repeatability (EV), reproducibility (AV), the interaction of operators and parts (IV), and part variation (PV) are calculated.

$$\begin{aligned} EV &= \nu\sqrt{\sigma^2_{EV}} \\ AV &= \nu\sqrt{\sigma^2_{AV}} \\ IV &= \nu\sqrt{\sigma^2_{IV}} \\ PV &= \nu\sqrt{\sigma^2_{PV}} \end{aligned}$$

When using the variance components method, the measure of gage repeatability and reproducibility has another component, the interaction term.

$$R\&R = \sqrt{(EV)^2 + (AV)^2 + (IV)^2}$$

Total variation is calculated similar to the average and range method.

$$TV = \sqrt{(R\&R)^2 + (PV)^2}$$

The results you get using the variance components method will differ slightly from those you get using the average and range method. This is because the variance components method is more precise, and the variance components method incorporates an interaction term in the measure of gage R&R.

As with the average and range method, the right-hand side of the report can be a percent of process variation or a percent of tolerance. $\%EV$, $\%AV$, $\%IV$, $\%R\&R$, and $\%PV$ are calculated similar to the average and range method.

The variance components method is more flexible than the average and range method in terms of the data that it can handle. Data for the average and range method should be balanced with the same number of parts and trials for each operator in the study. For example, if your study is composed of two operators, two trials, and ten parts, each operator should have 20 measurements. If the measurement for operator one, trial two, part three is missing, the average and range method cannot compute the gage measures. However, the variance components method can handle such missing data.

The average and range method also requires that a minimum number of parts be collected depending on the number of operators and the number of trials. Otherwise, the estimates will be imprecise. This is another situation where the variance components method can be used.

Note: The flexibility of the variance components method does not imply that you should not use locally recommended procedures for setting up and collecting data for gage studies.

Only a subset of the capabilities of PROC MIXED is used in the GAGE application. The procedure is capable of analyzing much more sophisticated statistical models. For example, you could fit an extended model to study the variability among several gages.

Creating a Data Set Outside the GAGE Application

Note: This section assumes you have some knowledge of creating SAS data sets.

Suppose your gage study data are stored in an external file, and you want to use the GAGE application but do not want to type in the data. How do you create a SAS data set that can be read by the GAGE application?

Table A.2 lists the SAS variables needed for the general information and the measurements in a GAGE data set.

Table A.2. GAGE Data Set Variables

Description	Variable			
	Name	**Type**	**Length**	**Values**
Test ID	TESTID	character	8	
Date	DATE	numeric	8	
Performed By	WHO	character	30	
Part No. & Name	PART	character	20	
Characteristics	CHAR	character	20	
Specification	SPEC	character	20	
Gage Name	GAGENAME	character	20	
Gage No.	GAGENO	character	20	
Gage Type	GAGETYPE	character	20	
Sigma Multiple	SPREAD	numeric	8	4, 5.15, 6
Analysis	PTYPE	character	1	T, V
Tolerance	TOL	numeric	8	
Operator (condition)	CONDITN	character	8	
Part	SAMPLE	numeric	8	1–15
Trial 1	TRIAL1	numeric	8	
Trial 2	TRIAL2	numeric	8	
Trial 3	TRIAL3	numeric	8	
Trial 4	TRIAL4	numeric	8	

A GAGE data set must have one observation for each combination of values of CONDITN and PART. You can have up to four operators (values of CONDITN), and each must be assigned a unique value. The MMDDYY8. format must be associated with the DATE variable. A PTYPE value of V indicates that you want a percent of process variation analysis. A PTYPE value of T indicates that you want a percent of tolerance analysis. The variable TOL must be assigned a tolerance value when PTYPE = T.

Return to the gasket thickness gage example described in the "Getting Started" section beginning on page 2060. Assume you are using the SAS System under Microsoft Windows, and the gasket thicknesses are stored in the external file c:\gage\gthick.dat. A partial listing of the data in gthick.dat is as follows:

```
Columns: 0----+----1----+----2----+----3
         George    1   0.65      0.6
         George    2   1.0       1.0
         George    3   0.85      0.8
         George    4   0.85      0.95
                   .
                   .
                   .
         Robert    7   0.95      0.95
         Robert    8   0.8       0.8
         Robert    9   1.05      1.05
         Robert   10   0.85      0.8
```

The following statements read the data into a GAGE SAS data set named SASUSER.GASKET. The LENGTH statement guarantees that each variable is represented in the data set. For the GAGE application to identify the data set as valid, the TYPE and ALTER data set options must be specified as shown.

```
data sasuser.gasket (type=_GRR alter=_GAGE);
   length testid $8 date 8 who $30 part char spec gagename
          gageno gagetype $20 spread 8 ptype $1 tol 8
          conditn $8 sample trial1 trial2 trial3 trial4 8;
   format date mmddyy8.;
   retain testid   'Gasket'          date     '17apr02'd
          who      'John Smith'      part     'Gasket'
          spec     '0.6-1.0 mm'      gagename 'Thickness'
          gageno   'X-2034'          gagetype '0-10 mm'
          spread   5.15              ptype    'T'
          tol      0.4;
   infile 'c:\gage\gthick.dat';
   input conditn $ 1-8 sample 10-11 trial1 15-18
         trial2 25-28;
run;
```

Figure A.26 is a partial listing of the SAS data set SASUSER.GASKET.

OBS	TESTID	DATE	WHO	PART	CHAR	SPEC	GAGENAME	GAGENO	GAGETYPE
1	Gasket	04/17/02	John Smith	Gasket		0.6-1.0 mm	Thickness	X-2034	0-10 mm
2	Gasket	04/17/02	John Smith	Gasket		0.6-1.0 mm	Thickness	X-2034	0-10 mm
3	Gasket	04/17/02	John Smith	Gasket		0.6-1.0 mm	Thickness	X-2034	0-10 mm
4	Gasket	04/17/02	John Smith	Gasket		0.6-1.0 mm	Thickness	X-2034	0-10 mm

OBS	SPREAD	PTYPE	TOL	CONDITN	SAMPLE	TRIAL1	TRIAL2	TRIAL3	TRIAL4
1	5.15	T	0.4	George	1	0.65	0.60	.	.
2	5.15	T	0.4	George	2	1.00	1.00	.	.
3	5.15	T	0.4	George	3	0.85	0.80	.	.
4	5.15	T	0.4	George	4	0.85	0.95	.	.
					.				
					.				
					.				

OBS	TESTID	DATE	WHO	PART	CHAR	SPEC	GAGENAME	GAGENO	GAGETYPE
27	Gasket	04/17/02	John Smith	Gasket		0.6-1.0 mm	Thickness	X-2034	0-10 mm
28	Gasket	04/17/02	John Smith	Gasket		0.6-1.0 mm	Thickness	X-2034	0-10 mm
29	Gasket	04/17/02	John Smith	Gasket		0.6-1.0 mm	Thickness	X-2034	0-10 mm
30	Gasket	04/17/02	John Smith	Gasket		0.6-1.0 mm	Thickness	X-2034	0-10 mm

OBS	SPREAD	PTYPE	TOL	CONDITN	SAMPLE	TRIAL1	TRIAL2	TRIAL3	TRIAL4
27	5.15	T	0.4	Robert	7	0.95	0.95	.	.
28	5.15	T	0.4	Robert	8	0.80	0.80	.	.
29	5.15	T	0.4	Robert	9	1.05	1.05	.	.
30	5.15	T	0.4	Robert	10	0.85	0.80	.	.

Figure A.26. The SAS Data Set SASUSER.GASKET

Extensibility of the Application

The GAGE application was not designed for any particular industry or company. Because many companies have their own techniques and guidelines for gage studies, the application is designed so that you can tailor it to suit your needs.

The interface to the GAGE application was implemented using FRAME entries in SAS/AF software and the SAS Screen Control Language (SCL). The FRAME entries and SCL source code are available in the gage catalog. FRAME entries provide a flexible environment for building graphical user interfaces. SCL is a programming language that enhances the capabilities of SAS/FSP software and SAS/AF software, including FRAME entries.

For further information on FRAME entries and SCL, refer to *SAS Component Language: Reference* and to *SAS Screen Control Language: Reference*.

Appendix B
The RELIABILITY Graphical Interface

An experimental graphical interface for the RELIABILITY procedure is implemented using FRAME entries in SAS/AF software. The application is available as a SAS/QC sample library program and is stored in the reliab catalog. (File extensions for SAS catalogs differ based on the operating system.)

Assume that you are using the SAS System under Microsoft Windows and that the SAS/QC sample library is stored in the c:\sas\qc\sample directory. (Check with your SAS site representative for the location of the SAS/QC sample library on your system.) You invoke the RELIABILITY application as follows:

1. First you must tell the SAS System where the catalog is stored:

```
libname rel 'c:\sas\qc\sample';
```

2. You then issue the following command from any SAS display manager window:

```
af c=rel.reliab.reliab.frame aws=no
```

The main application window appears, as shown in Figure B.1. You select a type of analysis from the main window. For example, you can select a probability plot by clicking the Probability Plot button.

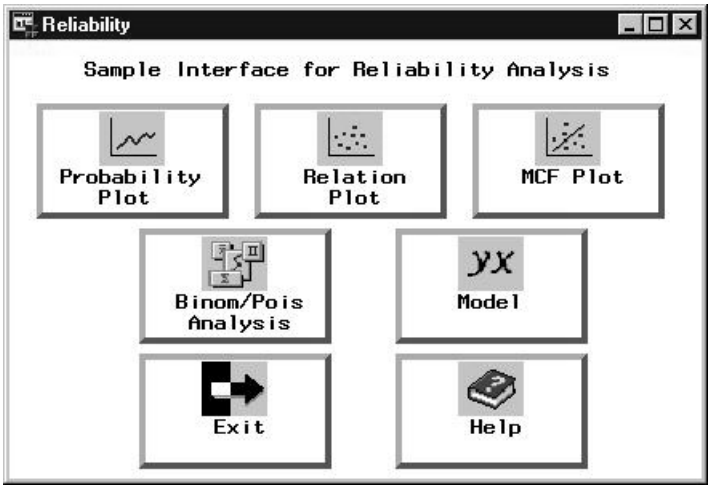


Figure B.1. Main Window

The next window to appear enables you to select the SAS data set that contains your data. You also specify a probability distribution for the probability plot and associated analysis. In Figure B.2, the data set WORK.FAN that contains the data for the engine fan example on page 1085 is selected, and the Weibull distribution is specified.

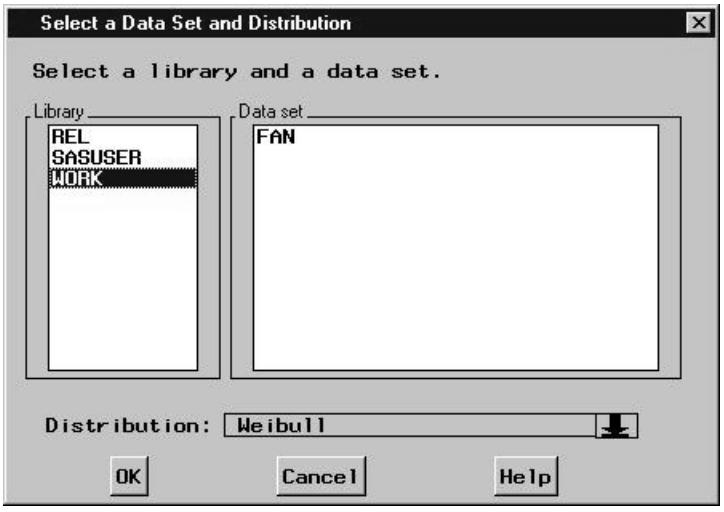


Figure B.2. Data Set and Distribution Window

Click the OK button, and the variable selection screen shown in Figure B.3 appears. The variable LIFETIME from the input data set is selected in Figure B.3 as the response variable, and the variable CENSOR is selected as the censoring indicator, with a value of 1 indicating censored lifetimes.

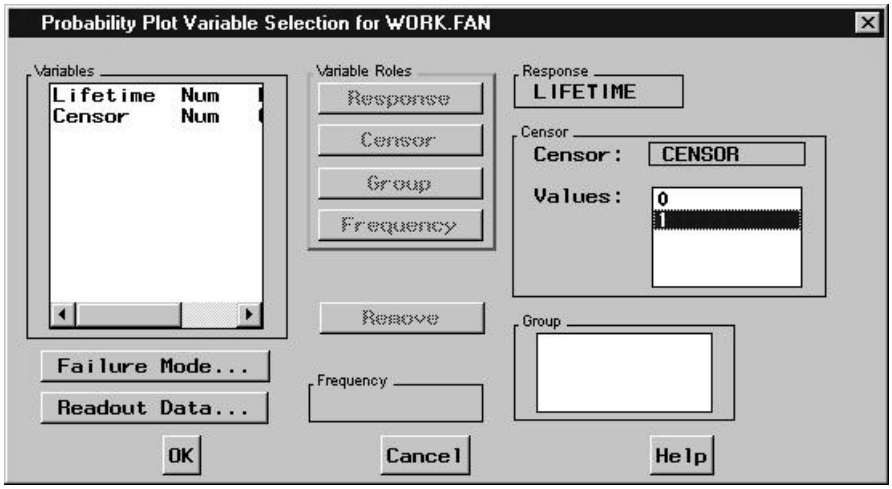


Figure B.3. Variable Selection Window

Clicking the OK button produces the probability plot window shown in Figure B.4. The RESULTS button enables you to view the tabular output from the RELIABILITY procedure.

You can choose procedure options and other analyses by selecting one of the menus at the top of the plot window. For example, you can specify additional plot options by selecting the plot menu, as shown in Figure B.5.

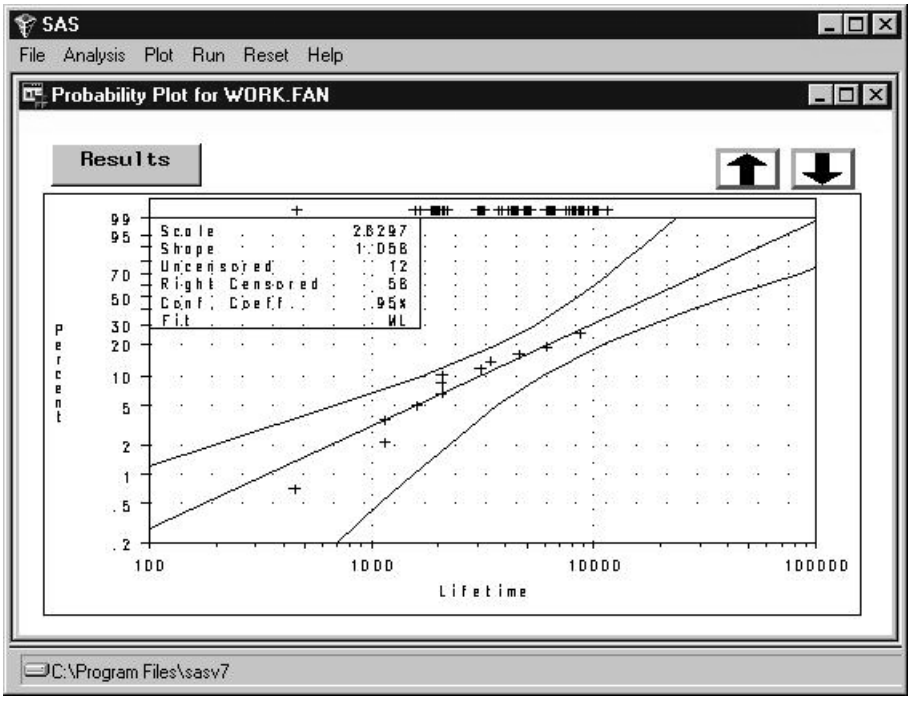


Figure B.4. Probability Plot Window

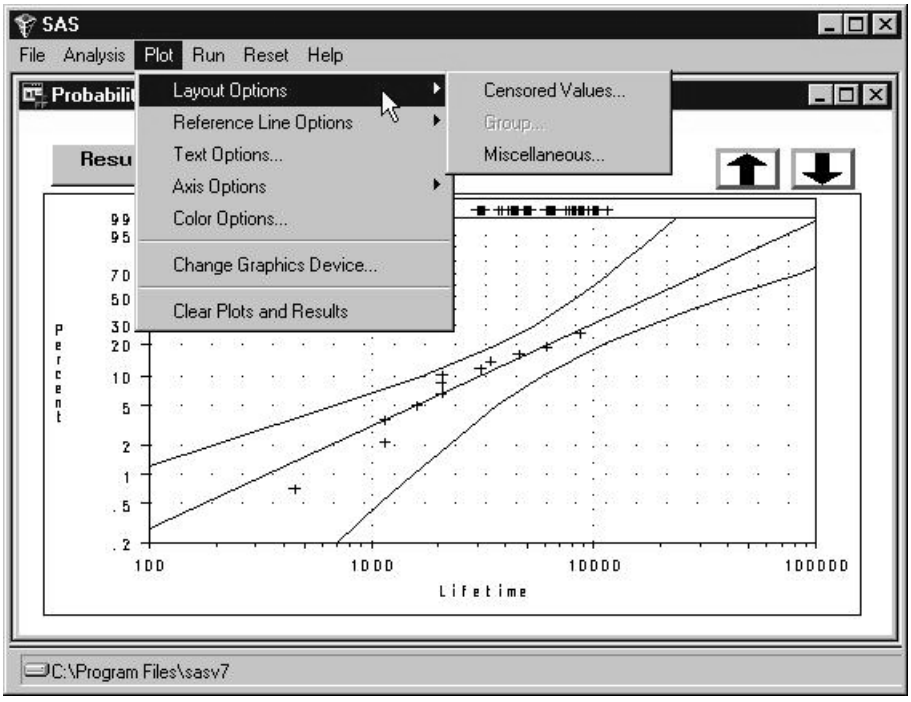


Figure B.5. Probability Plot Window

Appendix C
Functions

Appendix Contents

INTRODUCTION 2091

FUNCTION DESCRIPTIONS 2092
- AOQ2 Function 2092
- ASN2 Function 2093
- ATI2 Function 2095
- BAYESACT Call 2096
- C4 Function 2098
- CUSUMARL Function 2099
- D2 Function 2101
- D3 Function 2102
- EWMAARL Function 2103
- PROBACC2 Function 2104
- PROBBNML Function 2106
- PROBHYPR Function 2107
- PROBMED Function 2110
- STDMED Function 2111

DETAILS 2113
- Types of Sampling Plans 2113
- Evaluating Single-Sampling Plans 2113
- Evaluating Double-Sampling Plans 2115
- Deriving Control Chart Constants 2115

Appendix C
Functions

Introduction

SAS/QC software provides specialized DATA step functions for computations related to control chart analysis, for Bayes analysis of screening designs, and for sampling plan evaluation. You can use these functions in DATA step programming statements. The following lists summarize these functions:

Table C.1. Functions for Control Chart Analysis

Function	Description
C4	expected value c_4 of the standard deviation of a sample from a normal population with unit standard deviation
CUSUMARL	average run length for a cumulative sum control chart scheme
D2	expected value d_2 of the range of a sample from a normal population with unit standard deviation
D3	standard deviation d_3 of the range of a sample from a normal population with unit standard deviation
EWMAARL	average run length for an EWMA scheme
PROBMED	cumulative distribution function of sample median
STDMED	standard deviation of median of a standard normal sample

Table C.2. Function for Bayes Analysis of Screening Designs

Function	Description
BAYESACT	posterior probabilities of variance contamination

Table C.3. Functions for Sampling Plan Evaluation

Function	Description
AOQ2	average outgoing quality for double-sampling plan
ASN2	average sample number for double-sampling plan
ATI2	average total inspection for double-sampling plan
PROBACC2	acceptance probability for double-sampling plan

In addition, the PROBBNML and PROBHYPR functions, which are provided in Base SAS software, are useful when evaluating single-sampling plans.

The twelve SAS/QC functions, together with the PROBBNML and PROBHYPR functions, are described in the next section, "Function Descriptions." The "Details" section on page 2113, summarizes types of sampling plans and gives additional definitions.

Function Descriptions

This section describes the twelve SAS/QC functions and the related functions PROBBNML and PROBHYPR in alphabetical order.

AOQ2 Function

computes average outgoing quality for a double-sampling plan.

Syntax

AOQ2(*replacement*,$N, a_1, r_1, a_2, n_1, n_2, p$)

where

replacement	has the value `'REP'` or `'NOREP'`, respectively, depending on whether nonconforming items are replaced with conforming items.
N	is the lot size, where $N \geq 2$.
a_1	is the acceptance number for the first sample, where $a_1 \geq 0$.
r_1	is the rejection number for the first sample, where $r_1 > a_1 + 1$.
a_2	is the acceptance number for the second sample, where $a_2 \geq a_1$.
n_1	is the size of the first sample, where $n_1 \geq 1$ and $n_1 + n_2 \leq N$.
n_2	is the size of the second sample, where $n_2 \geq 1$ and $n_1 + n_2 \leq N$.
p	is the proportion of nonconforming items produced by the process, where $0 < p < 1$.

Description

The AOQ2 function returns the average outgoing quality for a Type B double-sampling plan in which nonconforming items are replaced with conforming items (*replacement* is `'REP'`) or not replaced (*replacement* is `'NOREP'`). For details on Type B double-sampling plans, see "Types of Sampling Plans" on page 2113.

For replacement, the average outgoing quality is

$$\text{AOQ} = \frac{pP_{a_1}(N - n_1) + pP_{a_2}(N - n_1 - n_2)}{N}$$

and for no replacement, the average outgoing quality is

$$\text{AOQ} = \frac{pP_{a_1}(N - n_1)}{N - n_1 p} + \frac{pP_{a_2}(N - n_1 - n_2)}{N - n_1 p - n_2 p}$$

where, in both situations,

$$
\begin{aligned}
P_{a_1} &= \sum_{d=0}^{a_1} f(d|n) \\
&= \text{probability of acceptance for first sample} \\
P_{a_2} &= \sum_{d=a_1+1}^{r_1-1} f(d|n_1)F(a_2-d|n_2) \\
&= \text{probability of acceptance for second sample}
\end{aligned}
$$

and

$$
\begin{aligned}
f(d|n) &= \binom{n}{d}p^d(1-p)^{n-d} \\
&= \text{binomial probability that the number of nonconforming items in a sample of size } n \text{ is exactly } d \\
F(a|n) &= \sum_{d=0}^{a} f(d|n) \\
&= \text{probability that the number of nonconforming items is less than or equal to } a
\end{aligned}
$$

Examples

The first set of statements results in a value of 0.0148099904. The second set of statements results in a value of 0.0144743043.

```
data;
   aoq=aoq2('norep',120,0,2,1,13,13,0.18);
   put aoq;
run;

data;
   aoq=aoq2('rep',120,0,2,1,13,13,0.18);
   put aoq;
run;
```

ASN2 Function

computes the average sample number for a double-sampling plan.

Syntax

ASN2(*mode*,$a_1, r_1, a_2, n_1, n_2, p$)

where

mode	identifies whether sampling is under full inspection (*mode* is `'FULL'`) or semicurtailed inspection (*mode* is `'SEMI'`).
a_1	is the acceptance number for the first sample, where $a_1 \geq 0$.
r_1	is the rejection number for the first sample, where $r_1 > a_1 + 1$.
a_2	is the acceptance number for the second sample, where $a_2 \geq a_1$.
n_1	is the size of the first sample, where $n_1 \geq 1$.
n_2	is the size of the second sample, where $n_2 \geq 1$.
p	is the proportion of nonconforming items produced by the process, where $0 < p < 1$.

Description

The ASN2 function returns the average sample number for a Type B double-sampling plan under full inspection (*mode* is `'FULL'`) or semicurtailed inspection (*mode* is `'SEMI'`). For details on Type B double-sampling plans, see "Types of Sampling Plans" on page 2113.

For full inspection, the average sample number is

$$\text{ASN} = n_1 + n_2[F(r_1 - 1|n_1) - F(a_1|n_1)]$$

and for semicurtailed inspection, the average sample number is

$$\text{ASN} = n_1 + \sum_{d=a_1+1}^{r_1-1} f(d|n_1)\left(n_2 F(a_2 - d|n_2) + \frac{r_2 - d}{p}[1 - F(r_2 - d|n_2 + 1)]\right)$$

where

$$\begin{aligned} f(d|n) &= \binom{n}{d} p^d (1-p)^{n-d} \\ &= \text{binomial probability that the number of nonconforming items in a sample of size } n \text{ is exactly } d \\ F(a|n) &= \sum_{d=0}^{a} f(d|n) \\ &= \text{probability that the number of nonconforming items is less than or equal to } a \end{aligned}$$

Examples

The first set of statements results in a value of 15.811418112. The second set of statements results in a value of 14.110408695.

```
data;
   asn=asn2('full',0,2,1,13,13,0.18);
   put asn;
run;

data;
   asn=asn2('semi',0,2,1,13,13,0.18);
   put asn;
run;
```

ATI2 Function

computes the average total inspection for a double-sampling plan.

Syntax

ATI2$(N, a_1, r_1, a_2, n_1, n_2, p)$

where

N	is the lot size, where $N \geq 2$.
a_1	is the acceptance number for the first sample, where $a_1 \geq 0$.
r_1	is the rejection number for the first sample, where $r_1 > a_1 + 1$.
a_2	is the acceptance number for the second sample, where $a_2 \geq a_1$.
n_1	is the size of the first sample, where $n_1 \geq 1$ and $n_1 + n_2 \leq N$.
n_2	is the size of the second sample, where $n_2 \geq 1$ and $n_1 + n_2 \leq N$.
p	is the proportion of nonconforming items produced by the process, where $0 < p < 1$.

Description

The ATI2 function returns the average total inspection for a Type B double-sampling plan. For details on Type B double-sampling plans, see "Types of Sampling Plans" on page 2113.

The average total inspection is

$$\text{ATI} = n_1 P_{a_1} + (n_1 + n_2) P_{a_2} + N(1 - P_{a_1} - P_{a_2})$$

where

$$\begin{aligned} P_{a_1} &= \sum_{d=0}^{a_1} f(d|n) \\ &= \text{probability of acceptance for first sample} \\ P_{a_2} &= \sum_{d=a_1+1}^{r_1-1} f(d|n_1) F(a_2 - d|n_2) \\ &= \text{probability of acceptance for second sample} \end{aligned}$$

and

$$\begin{aligned} f(d|n) &= \binom{n}{d} p^d (1-p)^{n-d} \\ &= \text{binomial probability that the number of nonconforming items in a sample of size } n \text{ is exactly } d \\ F(a|n) &= \sum_{d=0}^{a} f(d|n) \\ &= \text{probability that the number of nonconforming items is less than or equal to } a \end{aligned}$$

Examples

The following statements result in a value of 110.35046381:

```
data;
   ati=ati2(120,0,2,1,13,13,0.18);
   put ati;
run;
```

BAYESACT Call

computes posterior probabilities that observations are contaminated with a larger variance.

Syntax

CALL BAYESACT$(k, s, df, \alpha_1, \ldots, \alpha_n, y_1, \ldots, y_n, \beta_1, \ldots, \beta_n, p_0)$;

where

k	is the contamination coefficient, where $k \geq 1$.
s	is an independent estimate of σ, where $s \geq 0$.
df	is the number of degrees of freedom for s, where $df \geq 0$.
α_i	is the prior probability of contamination for the ith observation in the sample, where $i = 1, \ldots, n$ and n is the number of observations in the sample. Note that $0 \leq \alpha_i \leq 1$.
y_i	is the ith observation in the sample, where $i = 1, \ldots, n$ and n is the number of observations in the sample. When the BAYESACT call is used to perform a Bayes analysis of designs (see "Description" below), the y_is are estimates for effects.
β_i	is the variable that contains the returned posterior probability of contamination for the ith observation in the sample, where $i = 1, \ldots, n$ and n is the number of observations in the sample.
p_0	is the variable that contains the posterior probability that the sample is uncontaminated.

Description

The BAYESACT call computes posterior probabilities (β_i) that observations in a sample are *contaminated* with a larger variance than other observations and computes the posterior probability (p_0) that the entire sample is uncontaminated.

Specifically, the BAYESACT call assumes a normal random sample of n independent observations, with a mean of 0 (a centered sample) where some of the observations may have a larger variance than others:

$$\text{Var}(y_i) = \begin{cases} \sigma^2 & \text{with probability } 1-\alpha_i \\ k^2\sigma^2 & \text{with probability } \alpha_i \end{cases}$$

where $i = 1, \ldots, n$. The parameter k is called the *contamination coefficient*. The value of α_i is the *prior probability* of contamination for the ith observation. Based on the prior probability of contamination for each observation, the call gives the posterior probability of contamination for each observation and the posterior probability that the entire sample is uncontaminated.

Box and Meyer (1986) suggest computing posterior probabilities of contamination for the analysis of saturated orthogonal factorial designs. Although these designs give uncorrelated estimates for effects, the significance of effects cannot be tested in an analysis of variance since there are no degrees of freedom for error. Box and Meyer suggest computing posterior probabilities of contamination for the effect estimates. The prior probabilities (α_i) give the likelihood that an effect will be significant, and the contamination coefficient (k) gives a measure of how large the significant effect will be. Box and Meyer recommend using $\alpha = 0.2$ and $k = 10$, implying that about 1 in 5 effects will be about 10 times larger than the remaining effects. To adequately explore posterior probabilities, examine them over a range of values for prior probabilities and a range of contamination coefficients.

If an independent estimate of σ is unavailable (as is the case when the y_is are effects from a saturated orthogonal design), use 0 for s and df in the BAYESACT call. Otherwise, the call assumes s is proportional to the square root of a χ^2 random variable with df degrees of freedom. For example, if the y_is are estimated effects from an orthogonal design that is not saturated, then use the BAYESACT call with s equal to the estimated standard error of the estimates and df equal to the degrees of freedom for error.

From Bayes' theorem, the posterior probability that y_i is contaminated is

$$\beta_i(\sigma) = \frac{\alpha_i f(y_i; 0, k^2\sigma^2)}{\alpha_i f(y_i; 0, k^2\sigma^2) + (1-\alpha_i) f(y_i; 0, \sigma^2)}$$

for a given value of σ, where $f(x; \mu, \sigma)$ is the density of a normal distribution with mean μ and variance σ^2.

The probability that the sample is uncontaminated is

$$p = \prod_{i=1}^{n} (1 - \beta_i(\sigma))$$

Posterior probabilities that are independent of σ are derived by integrating $\beta_i(\sigma)$ and p over a noninformative prior for σ. If an estimate of σ is available (when $df > 0$), it is appropriately incorporated. Refer to Box and Meyer (1986) for details.

Examples

The statements

```
data;
   retain post1-post7 postnone;
   call bayesact(10,0,0,
          0.2,   0.2,   0.2,   0.2,   0.2,     0.2,   0.2,
      -5.4375,1.3875,8.2875,0.2625,1.7125,-11.4125,1.5875,
        post1, post2, post3, post4, post5,   post6, post7,
      postnone);
run;
```

return the following posterior probabilities:

```
POST1      0.42108
POST2      0.037412
POST3      0.53438
POST4      0.024679
POST5      0.050294
POST6      0.64329
POST7      0.044408
POSTNONE   0.28621
```

The probability that the sample is uncontaminated is 0.28621. A situation where this BAYESACT call would be appropriate is a saturated 2^7 design in 8 runs, where the estimates for main effects are as shown in the function above (-5.4375, 1.3875, . . . , 1.5875).

C4 Function

computes the expected value of the standard deviation of n independent normal random variables.

Syntax

C4(n)

where n is the sample size, with $n \geq 2$.

Description

The C4 function returns the expected value of the standard deviation of n independent, normally distributed random variables with the same mean and with standard deviation of 1. This expected value is referred to as the control chart constant c_4.

The value c_4 is calculated as

$$c_4 = \frac{\Gamma(\frac{n}{2})\sqrt{2/(n-1)}}{\Gamma(\frac{n-1}{2})}$$

where $\Gamma(\cdot)$ is the gamma function. As n grows, c_4 is asymptotically equal to $(4n-4)/(4n-3)$.

For more information, refer to the *ASQC Glossary and Tables for Statistical Quality Control*, the *ASTM Manual on Presentation of Data and Control Chart Analysis*, Montgomery (1996), and Wadsworth and others (1986).

In other chapters, c_4 is written as $c_4(n)$ to emphasize the dependence on n.

You can use the constant c_4 to calculate an unbiased estimate $(\hat{\sigma})$ of the standard deviation σ of a normal distribution from the sample standard deviation of n observations:

$$\hat{\sigma} = (\text{sample standard deviation})/c_4$$

where the sample standard deviation is calculated using $n-1$ in the denominator. In the SHEWHART procedure, c_4 is used to calculate control limits for s charts, and it is used in the estimation of the process standard deviation based on subgroup standard deviations.

Examples

The following statements result in a value of 0.939985603:

```
data;
   constant=c4(5);
   put constant;
run;
```

CUSUMARL Function

computes the average run length for a cumulative sum control chart scheme.

Syntax

CUSUMARL(*type*,δ, h, k <,*headstart*>)

where

type	indicates a one-sided or two-sided scheme. Valid values are 'ONESIDED' or 'O' for a one-sided scheme, and 'TWOSIDED' or 'T' for a two-sided scheme.
δ	is the shift to be detected, expressed as a multiple of the process standard deviation (σ).
h	is the decision interval (one-sided scheme) or the vertical distance between the origin and the upper arm of the V-mask (two-sided scheme), each time expressed as a positive value in standard units (a multiple of $\sigma/\sqrt{n}$, where n is the subgroup sample size).
k	is the reference value (one-sided scheme) or the slope of the lower arm of the V-mask (two-sided scheme), each time expressed as a positive value in standard units (a multiple of $\sigma/\sqrt{n}$, where n is the subgroup sample size).
headstart	is the headstart value (optional) expressed in standard units (a multiple of $\sigma/\sqrt{n}$, where n is the subgroup sample size). The default *headstart* is zero. For details, refer to Lucas and Crosier (1982).

Description

The CUSUMARL function returns the average run length of one-sided and two-sided cumulative sum schemes with parameters as described above. The notation is consistent with that used in the CUSUM procedure.

For a one-sided scheme, the average run length is calculated using the integral equation method (with 24 Gaussian points) described by Goel and Wu (1971) and Lucas and Crosier (1982).

For a two-sided scheme with no *headstart*, the average run length (ARL) is calculated using the fact that

$$(\text{ARL})^{-1} = (\text{ARL}_+)^{-1} + (\text{ARL}_-)^{-1}$$

where ARL_+ and ARL_- denote the average run lengths of the equivalent one-sided schemes for detecting a shift of the same magnitude in the positive direction and in the negative direction, respectively.

For a two-sided scheme with a nonzero *headstart*, the ARL is calculated by combining average run lengths for one-sided schemes as described in Appendix A.1 of Lucas and Crosier (1982, 204).

For a specified shift δ, you can use the CUSUMARL function to design a cusum scheme by first calculating average run lengths for a range of values of h and k and then choosing the combination of h and k that yields a desired average run length.

You can also use the CUSUMARL function to interpolate published tables of average run lengths.

Examples

The following three sets of statements result in the values 4.1500826715, 4.1500836225, and 4.1061588131, respectively.

```
data;
   arl=cusumarl('twosided',2.5,8,0.25);
   put arl;
run;

data;
   arl=cusumarl('onesided',2.5,8,0.25);
   put arl;
run;

data;
   arl=cusumarl('o',2.5,8,0.25,0.1);
   put arl;
run;
```

D2 Function

computes the expected value of the sample range.

Syntax

D2(n)

where n is the sample size, with $2 \leq n \leq 25$.

Description

The D2 function returns the expected value of the sample range of n independent, normally distributed random variables with the same mean and a standard deviation of 1. This expected value is referred to as the control chart constant d_2. The values returned by the D2 function are accurate to ten decimal places.

The value d_2 can be expressed as

$$d_2 = \int_{-\infty}^{\infty} [1 - (1 - \Phi(x))^n - (\Phi(x))^n] \; dx$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function. Refer to Tippett (1925). In other chapters, d_2 is written as $d_2(n)$ to emphasize the dependence on n.

In the SHEWHART procedure, d_2 is used to calculate control limits for r charts, and it is used in the estimation of the process standard deviation based on subgroup ranges. Also refer to the *ASQC Glossary and Tables for Statistical Quality Control*, the *ASTM Manual on Presentation of Data and Control Chart Analysis*, Kume (1985), Montgomery (1996), and Wadsworth and others (1986).

You can use the constant d_2 to calculate an unbiased estimate $(\hat{\sigma})$ of the standard deviation σ of a normal distribution from the sample range of n observations:

$$\hat{\sigma} = (\text{sample range})/d_2$$

Note that the statistical efficiency of this estimate relative to that of the sample standard deviation decreases as n increases.

Examples

The following statements result in a value of 2.3259289473:

```
data;
   constant=d2(5);
   put constant;
run;
```

D3 Function

computes the standard deviation of the range of n independent normal random variables.

Syntax

D3(n)

where n is the sample size, with $2 \leq n \leq 25$.

Description

The D3 function returns the standard deviation of the range of n independent, normally distributed random variables with the same mean and with unit standard deviation. The standard deviation returned is referred to as the control chart constant d_3. The values returned by the D3 function are accurate to ten decimal places.

The value d_3 can be expressed as

$$d_3 = \sqrt{2\int_{-\infty}^{\infty}\int_{-\infty}^{y} f(x,y)\,dx\,dy - d_2^2}$$

where

$$f(x,y) = 1 - (\Phi(y))^n - (1-\Phi(x))^n + (\Phi(y) - \Phi(x))^n$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function and d_2 is the expected range. Refer to Tippett (1925).

In other chapters d_3 is written as $d_3(n)$ to emphasize the dependence on n.

In the SHEWHART procedure, d_3 is used to calculate control limits for r charts, and it is used in the estimation of the process standard deviation based on subgroup ranges.

For more information, refer to the *ASQC Glossary and Tables for Statistical Quality Control*, the *ASTM Manual on Presentation of Data and Control Chart Analysis*, Montgomery (1996), and Wadsworth and others (1986).

You can use the constant d_3 to calculate an unbiased estimate $(\hat{\sigma})$ of the standard deviation σ_R of the range of a sample of n normally distributed observations from the sample range of n observations:

$$\hat{\sigma}_R = (\text{sample range})(d_3/d_2)$$

You can use the D2 function to calculate d_2.

Examples

The following statements result in a value of 0.8640819411:

```
data;
   constant=d3(5);
   put constant;
run;
```

EWMAARL Function

computes the average run length for an exponentially weighted moving average.

Syntax

EWMAARL(δ, r, k)

where

δ	is the shift to be detected, expressed as a multiple of the process standard deviation (σ), where $\delta \geq 0$.
r	is the weight factor for the current subgroup mean in the EWMA, where $0 < r \leq 1$. If $r = 1$, the EWMAARL function returns the average run length for a Shewhart chart for means. Refer to Wadsworth and others (1986). If $r \leq 0.05$, $k \geq 3$, and $\delta < 0.10$, the algorithm used is unstable. However, note that the EWMA behaves like a cusum when $r \rightarrow 0$, and in this case the CUSUMARL function is applicable.
k	is the multiple of σ used to define the control limits, where $k \geq 0$. Typically $k = 3$.

Description

The EWMAARL function computes the average run length for an exponentially weighted moving average (EWMA) scheme using the method of Crowder (1987a,b). The notation used in the preceding list is consistent with that used in the MACONTROL procedure.

For a specified shift δ, you can use the EWMAARL function to design an exponentially weighted moving average scheme by first calculating average run lengths for a range of values of r and k and then choosing the combination of r and k that yields a desired average run length.

Examples

The following statements specify a shift of 1σ, a weight factor of 0.25, and 3σ control limits. The EWMAARL function returns an average run length of 11.154267016.

```
data;
   arl=ewmaarl(1.00,0.25,3.0);
   put arl;
run;
```

PROBACC2 Function

computes the acceptance probability for a double-sampling plan.

Syntax

PROBACC2$(a_1, r_1, a_2, n_1, n_2, D, N)$

PROBACC2$(a_1, r_1, a_2, n_1, n_2, p)$

where

a_1	is the acceptance number for the first sample, where $a_1 \geq 0$.
r_1	is the rejection number for the first sample, where $r_1 > a_1 + 1$.
a_2	is the acceptance number for the second sample, where $a_2 > a_1$.
n_1	is the size of the first sample, where $n_1 \geq 1$ and $n_1 + n_2 \leq N$.
n_2	is the size of the second sample, where $n_2 \geq 1$ and $n_1 + n_2 \leq N$.
D	is the number of nonconforming items in the lot, where $0 \leq D \leq N$.
N	is the lot size, where $N \geq 2$.
p	is the proportion of nonconforming items produced by the process, where $0 < p < 1$.

Description

The PROBACC2 function returns the acceptance probability for a double-sampling plan of Type A if you specify the parameters D and N, and it returns the acceptance probability for a double-sampling plan of Type B if you specify the parameter p. For

details on Type A and Type B double-sampling plans, see "Types of Sampling Plans" on page 2113.

For either type of sampling plan, the acceptance probability is calculated as

$$P_{a_1} + P_{a_2}$$

where

$$\begin{aligned} P_{a_1} &= \sum_{d=0}^{a_1} f(d|n) \\ &= \text{probability of acceptance for first sample} \\ P_{a_2} &= \sum_{d=a_1+1}^{r_1-1} f(d|n_1)F(a_2-d|n_2) \\ &= \text{probability of acceptance for second sample} \end{aligned}$$

and

$$\begin{aligned} f(d|n) &= \binom{n}{d}p^d(1-p)^{n-d} \\ &= \text{binomial probability that the number of nonconforming items in a sample of size } n \text{ is exactly } d \\ F(a|n) &= \sum_{d=0}^{a} f(d|n) \\ &= \text{probability that the number of nonconforming items is less than or equal to } a \end{aligned}$$

These probabilities are determined from either the hypergeometric distribution (Type A sampling) or the binomial distribution (Type B sampling).

Examples

The first set of statements results in a value of 0.2396723824. The second set of statements results in a value of 0.0921738126.

```
data;
   prob=probacc2(1,4,3,50,100,10,200);
   put prob;
run;

data;
   prob=probacc2(0,2,1,13,13,0.18);
   put prob;
run;
```

PROBBNML Function

computes the probability that an observation from a binomial(n, p) distribution will be less than or equal to m.

Syntax

PROBBNML(p, n, m)

where

p	is the probability of success for the binomial distribution, where $0 \leq p \leq 1$. In terms of acceptance sampling, p is the probability of selecting a nonconforming item.
n	is the number of independent Bernoulli trials in the binomial distribution, where $n \geq 1$. In terms of acceptance sampling, n is the number of items in the sample.
m	is the number of successes, where $0 \leq m \leq n$. In terms of acceptance sampling, m is the number of nonconforming items.

Description

The PROBBNML function returns the probability that an observation from a binomial distribution (with parameters n and p) is less than or equal to m. To compute the probability that an observation is equal to a given value m, compute the difference of two values for the cumulative binomial distribution.

In terms of acceptance sampling, the function returns the probability of finding m or fewer nonconforming items in a sample of n items, where the probability of a nonconforming item is p. To find the probability that the sample contains exactly m nonconforming items, compute the difference between PROBBNML(p, n, m) and PROBBNML$(p, n, m-1)$.

In addition to using the PROBBNML function to return the probability of acceptance, the function can be used in calculations for average sample number, average outgoing quality, and average total inspection in Type B single-sampling. See "Evaluating Single-Sampling Plans" on page 2113 for details.

The PROBBNML function computes

$$\sum_{j=0}^{m} \binom{n}{j} p^j (1-p)^{n-j}$$

where m, n, and p are defined in the preceding list.

Examples

The following statements compute the probability that an observation from a binomial distribution with $p = 0.05$ and $n = 10$ is less than or equal to 4:

```
data;
   prob=probbnml(0.05,10,4);
   put prob;
run;
```

These statements result in the value 0.9999363102. In terms of acceptance sampling, for a sample of size 10 where the probability of a nonconforming item is 0.05, the probability of finding 4 or fewer nonconforming items is 0.9999363102.

The following statements compute the probability that an observation from a binomial distribution with $p = 0.05$ and $n = 10$ is exactly 4:

```
data;
   p=probbnml(0.05,10,4) - probbnml(0.05,10,3);
   put p;
run;
```

These statements result in the value 0.0009648081.

For additional information on probability functions, refer to *SAS Language Reference: Dictionary*.

PROBHYPR Function

computes the probability that an observation from a hypergeometric distribution is less than or equal to x.

Syntax

PROBHYPR$(N, K, n, x <, r >)$

where

N is the population size for a hypergeometric distribution. In terms of acceptance sampling, N is the lot size.

K is the number of items in the category of interest in the population. In terms of acceptance sampling, K is the number of nonconforming items in a lot.

n is the sample size for a hypergeometric distribution. In terms of acceptance sampling, n is the sample size.

x is the number of items from the category of interest in the sample. In terms of acceptance sampling, x is the number of nonconforming items in the sample.

r is optional and gives the odds ratio for the extended hypergeometric distribution. For the standard hypergeometric distribution, $r = 1$; this value is the default. In acceptance sampling, typically $r = 1$.

Restrictions on items in the syntax are given in the following equations:

$1 \leq N$
$0 \leq K \leq N$
$0 \leq n \leq N$
$\max(0, K + n - N) \leq x \leq \min(K, n)$
N, K, n and x are integers

Description

The PROBHYPR function returns the probability that an observation from an extended hypergeometric distribution with parameters N, K and n and an odds ratio of r is less than or equal to x. The default for r is 1 and leads to the usual hypergeometric distribution.

In terms of acceptance sampling, if $r = 1$, the PROBHYPR function gives the probability of x or fewer nonconforming items in a sample of size n taken from a lot containing N items, K of which are nonconforming, when sampling is done without replacement. Typically $r = 1$ in acceptance sampling.

For example, suppose an urn contains red and white balls, and you are interested in the probability of selecting a white ball. If $r = 1$, the function returns the probability of selecting x white balls when given the population size (number of balls in the urn), sample size (number of balls taken from the urn), and number of white balls in the population (urn).

If, however, the probability of selecting a white ball differs from the probability of selecting a red ball, then $r \neq 1$. Suppose an urn contains one white ball and one red ball, and the probability of choosing the red ball is higher than the probability of choosing the white ball. This might occur if the red ball were larger than the white ball, for example. Given the probabilities of choosing a red ball and a white ball when an urn contains one of each, you calculate r and use the value in the PROBHYPR function. Returning to the case where an urn contains many balls with $r \neq 1$, the function gives the probability of selecting x white balls when given the number of balls in the urn, the number of balls taken from the urn, the number of white balls in the urn, and the relative probability of selecting a white ball or a red ball.

The PROBHYPR function is used to evaluate Type A single-sampling plans. See "Evaluating Single-Sampling Plans" on page 2113 for details.

If $r = 1$ (the default), the PROBHYPR function calculates probabilities from the usual hypergeometric distribution:

$$\Pr[X \leq x] = \sum_{i=0}^{x} P_i$$

where

$$P_i = \begin{cases} \dfrac{\binom{K}{i}\binom{N-K}{n-i}}{\binom{N}{n}} & \text{if } \max(0, K+n-N) \le i \le \min(K, n) \\ 0 & \text{otherwise} \end{cases}$$

The PROBHYPR function accepts values other than 1 for r, and in these cases, it calculates the probability for the extended hypergeometric distribution:

$$\Pr[X_1 \le x | X_1 + X_2 = n] = \sum_{i=o}^{x} P_i$$

where

$$P_i = \begin{cases} \dfrac{\binom{K}{i}\binom{N-K}{n-i} r^j}{\sum_{j=0}^{n}\binom{K}{j}\binom{N-K}{n-j} r^j} & \text{if } \max(0, K+n-N) \le i \le \min(K, n) \\ 0 & \text{otherwise} \end{cases}$$

where

X_1	is binomially distributed with parameters K and p_1.
X_2	is binomially distributed with parameters N-K and p_2.
q_1	$= 1 - p_1$
q_2	$= 1 - p_2$
r	$= (p_1 q_2)/(p_2 q_1)$

For details on the extended hypergeometric distribution, refer to Johnson and Kotz (1969).

Examples

Suppose you take a sample of size 10 (without replacement) from an urn that contains 200 balls, 50 of which are white. The remaining 150 balls are red. The following statements calculate the probability that your sample contains 2 or fewer white balls:

```
data;
   y=probhypr(200,50,10,2);
   put y;
run;
```

These statements result in a value of 0.5236734081. Now, suppose the probability of selecting a red ball does not equal the probability of selecting a white ball. Specifically, suppose the probability of choosing a red ball is $p_2 = 0.4$ and the probability of choosing a white ball is $p_1 = 0.2$. Calculate r as

$$r = \frac{p_1 q_2}{p_2 q_1} = \frac{(0.2)(0.6)}{(0.4)(0.8)} = 0.375$$

With $r = 0.375$, the probability of choosing 2 or fewer white balls from an urn that contains 200 balls, 50 of which are white, is calculated using the following statements:

```
data;
   y=probhypr(200,50,10,2,0.375);
   put y;
run;
```

These statements return a value of 0.9053936127. See "Evaluating Single-Sampling Plans" on page 2113 for another example.

For additional information on probability functions, refer to *SAS Language Reference: Dictionary*.

PROBMED Function

computes cumulative probabilities for the sample median.

Syntax

PROBMED(n, x)

where

n	is the sample size.
x	is the point of interest; that is, the PROBMED function calculates the probability that the median is less than or equal to x.

Description

The PROBMED function computes the probability that the sample median is less than or equal to x for a sample of n independent, standard normal random variables (mean 0, variance 1).

Let n represent the sample size and $X_{(i)}$ represent the ith order statistic. Then, when n is odd, the function calculates

$$\Pr[X_{((n+1)/2)} \leq x] = \mathrm{I}_{\Phi(x)}\left(\frac{n+1}{2}, \frac{n+1}{2}\right)$$

where

$$\mathrm{I}_p(a,b) = \frac{1}{\mathrm{B}(a,b)} \int_0^p t^{a-1}(1-t)^{b-1}\, dt$$

and $\mathrm{B}(a,b) = \Gamma(a)\Gamma(b)/\Gamma(a+b)$, where $\Gamma(\cdot)$ is the gamma function. If n is even, the PROBMED function calculates

$$\Pr\left[\frac{X_{(n/2)} + X_{((n/2)+1)}}{2} \leq x\right] =$$

$$\frac{2}{\mathrm{B}(\frac{n}{2},\frac{n}{2})} \int_{-\infty}^{x} \left\{[1-\Phi(u)]^{n/2} - [1-\Phi(2x-u)]^{n/2}\right\} [\Phi(u)]^{(n/2)-1}\, \phi(u)\, du$$

where $\mathrm{B}(n/2, n/2) = [\Gamma(n/2)]^2/\Gamma(n)$ and $\Phi(\cdot)$ and $\phi(\cdot)$ are the standard normal cumulative distribution function and density function, respectively.

For more information, refer to David (1981).

Examples

The statements

```
data;
   b=probmed(5,-0.1);
   put b;
run;
```

result in a value of 0.4256380897.

STDMED Function

computes the standard deviation of a sample median.

Syntax

STDMED(n)

where n is the sample size.

Description

The STDMED function gives the standard deviation of the median of a normally distributed sample with a mean of 0 and a variance of 1. This function gives the standard error used to determine the width of the control limits for charts produced by the MCHART and MRCHART statements in PROC SHEWHART.

Let n represent the sample size and $X_{(i)}$ represent the ith order statistic. Then, when n is odd, the STDMED function calculates $\sqrt{\mathrm{Var}(X_{((n+1)/2)})}$, where

$$\mathrm{Var}(X_{((n+1)/2)}) = \frac{1}{\mathrm{B}\left(\frac{n+1}{2},\frac{n+1}{2}\right)} \int_{-\infty}^{\infty} x^2 [\Phi(x)]^{(n-1)/2} [1-\Phi(x)]^{(n-1)/2}\, \phi(x)\, dx$$

where $\mathrm{B}(a,b) = \Gamma(a)\Gamma(b)/\Gamma(a+b)$ and $\Gamma(\cdot)$ is the gamma function, $\Phi(\cdot)$ is the standard normal cumulative distribution function, and $\phi(\cdot)$ is the corresponding density function.

If n is even, the function calculates the square root of the following:

$$\mathrm{Var}\left[\frac{X_{(n/2)} + X_{((n/2)+1)}}{2}\right] =$$

$$(1/4)\left[\mathrm{E}(X^2_{(n/2)}) + \mathrm{E}(X^2_{((n/2)+1)}) + 2\mathrm{E}(X_{(n/2)}X_{((n/2)+1)})\right]$$

where

$$\mathrm{E}(X^2_{(n/2)}) = \frac{2}{\mathrm{B}\left(\frac{n}{2},\frac{n}{2}\right)}\int_{-\infty}^{\infty} x^2[\Phi(x)]^{(n/2)-1}[1-\Phi(x)]^{n/2}\,\phi(x)\,dx$$

$$\mathrm{E}(X^2_{((n/2)+1)}) = \frac{2}{\mathrm{B}\left(\frac{n}{2},\frac{n}{2}\right)}\int_{-\infty}^{\infty} x^2[\Phi(x)]^{n/2}[1-\Phi(x)]^{(n/2)-1}\,\phi(x)\,dx$$

$$\mathrm{E}(X_{(n/2)}X_{((n/2)+1)}) = \frac{n}{\mathrm{B}\left(\frac{n}{2},\frac{n}{2}\right)}\int_{-\infty}^{\infty}\int_{-\infty}^{y} xy[\Phi(x)]^{(n/2)-1}[1-\Phi(y)]^{(n/2)-1}\,\phi(x)\,\phi(y)\,dx\,dy$$

For more details, refer to David (1981), Kendall and Stuart (1977, 252), and Sarhan and Greenberg (1962).

Examples

These statements use a loop to calculate the standard deviation of the median for sample sizes from 6 to 12:

```
data;
   do n=6 to 12;
      s=stdmed(n);
      put s;
      output;
   end;
run;
```

The statements produce these values:

```
0.4634033519
0.4587448763
0.410098592
0.4075552495
0.3719226208
0.3703544701
0.3428063408
```

Details

Types of Sampling Plans

In single sampling, a random sample of n items is selected from a lot of size N. If the number d of nonconforming (defective) items found in the sample is less than or equal to an acceptance number c, the lot is accepted. Otherwise, the lot is rejected.

In double sampling, a sample of size n_1 is drawn from the lot, and the number d_1 of nonconforming items is counted. If d_1 is less than or equal to an acceptance number a_1, the lot is accepted, and if d_1 is greater than or equal to a rejection number r_1, the lot is rejected. Otherwise, if $a_1 < d_1 < r_1$, a second sample of size n_2 is taken, and the number of nonconforming items d_2 is counted. Then if $d_1 + d_2$ is less than or equal to an acceptance number a_2, the lot is accepted, and if $d_1 + d_2$ is greater than or equal to a rejection number $r_2 = a_2 + 1$, the lot is rejected. This notation follows that of Schilling (1982). Note that some authors, including Montgomery (1996), define the first rejection number using a strict inequality.

In *Type A sampling*, the sample is intended to represent a single, finite-sized lot, and the characteristics of the sampling plan depend on D, the number of nonconforming items in the lot, as well as N, n, and c.

In *Type B sampling*, the sample is intended to represent a series of lots (or the lot size is effectively infinite), and the characteristics of the sampling plan depend on p, the proportion of nonconforming items produced by the process, as well as n and c.

A hypergeometric model is appropriate for Type A sampling, and a binomial model is appropriate for Type B sampling.

Evaluating Single-Sampling Plans

You can use the Base SAS functions PROBBNML and PROBHYPR to evaluate single-sampling plans. Measures of the performance of single-sampling plans include

- the probability of acceptance P_a
- the average sample number ASN
- the average outgoing quality AOQ
- the average total inspection ATI

Probability of Acceptance

Since P_a is the probability of finding c or fewer defectives in the sample, you can calculate the acceptance probability using the function PROBHYPR(N, D, n, c) for Type A sampling and the function PROBBNML(p, n, c) for Type B sampling.

For example, the following statements calculate P_a for the plan $n = 20$, $c = 1$ when sampling from a single lot of size $N = 120$ that contains $D = 22$ nonconforming items, resulting in a value of 0.0762970752:

```
data;
   prob=probhypr(120,22,20,1);
   put prob;
run;
```

Similarly, the following statements calculate P_a for the plan $n = 20$, $c = 1$ when sampling from a series of lots for which the proportion of nonconforming items is $p = 0.18$, resulting in a value of 0.1018322793:

```
data;
   prob=probbnml(0.18,20,1);
   put prob;
run;
```

Other Measures of Performance

The measures ASN, AOQ, and ATI are meaningful only for Type B sampling and can be calculated using the PROBBNML function. For reference, the following equations are provided.

Average sample number: Following the notation of Schilling (1982), let $F(c|n)$ denote the probability of finding c or fewer nonconforming items in a sample of size n. Note that $F(c|n)$ is equivalent to PROBBNML(p, n, c). Then, depending on the mode of inspection, the average sample number can be expressed as shown in the following table:

Mode of Inspection	ASN
Full	n
Semicurtailed	$nF(c\|n) + \dfrac{(c+1)(1-F(c+1\|n+1))}{p}$
Fully curtailed	$\dfrac{(n-c)F(c\|n+1)}{1-p} + \dfrac{(c+1)(1-F(c+1\|n+1))}{p}$

Average outgoing quality can be expressed as

$$\text{AOQ} = \frac{p(N-n)F(c|n)}{N}$$

if the nonconforming items found are replaced with conforming items, and as

$$\text{AOQ} = \frac{p(N-n)F(c|n)}{N-np}$$

if the nonconforming items found are not replaced.

Average total inspection can be expressed as

$$\text{ATI} = n + (1 - F(c|n))(N-n)$$

Evaluating Double-Sampling Plans

The following list gives some measures for double-sampling plans. The formula for each measure is given in the section describing the corresponding function.

- the probability of acceptance, P_a, calculated with the PROBACC2 function
- the average sample number, ASN, calculated with the ASN2 function
- the average outgoing quality, AOQ, calculated with the AOQ2 function
- the average total inspection, ATI, calculated with the ATI2 function

Deriving Control Chart Constants

You can use the functions D2, D3, and C4 to calculate standard control chart constants that are derived from d_2, d_3 and c_4. For reference, the following equations for some of these constants are provided:

$$
\begin{aligned}
A_2 &= k/(d_2\sqrt{n}) \\
A_3 &= k/(c_4\sqrt{n}) \\
B_3 &= \max(0, 1-(k/c_4)\sqrt{1-c_4^2}\,) \\
B_4 &= 1+(k/c_4)\sqrt{1-c_4^2} \\
B_5 &= \max(0, c_4 - k\sqrt{1-c_4^2}) \\
B_6 &= c_4 + k\sqrt{1-c_4^2} \\
c_5 &= \sqrt{1-c_4^2} \\
D_1 &= \max(0, d_2 - kd_3) \\
D_2 &= d_2 + kd_3 \\
D_3 &= \max(0, 1 - kd_3/d_2) \\
D_4 &= 1 + kd_3/d_2 \\
E_2 &= k/d_2 \\
E_3 &= k/c_4
\end{aligned}
$$

In the preceding equations, k is the multiple of standard error ($k = 3$ in the case of 3σ limits), and n is the subgroup sample size. The use of these control chart constants is discussed in the *ASQC Glossary and Tables for Statistical Quality Control*, the *ASTM Manual on Presentation of Data and Control Chart Analysis*, Montgomery (1996), and Wadsworth and others (1986).

Although you do not ordinarily need to calculate control chart constants when using the SHEWHART procedure, you may find the D2, D3, and C4 functions useful for creating LIMITS= data sets that contain control limits to be read by the SHEWHART procedure.

Appendix D
Special Fonts in SAS/QC Software

Appendix Contents

INTRODUCTION 2119
FONT SELECTION 2119

Appendix D
Special Fonts in SAS/QC Software

Introduction

Four special graphics fonts named QCFONT1, QCFONT2, QCFONT3, and QCFONT4 are provided with SAS/QC software. You can use these fonts to display symbols such as $\bar{X}$, which are commonly encountered in statistical quality improvement applications, in the titles and footnotes of displays created with high-resolution graphics devices.

Font Selection

Each of the special fonts matches a particular SAS/GRAPH font, as indicated by the following table:

Special Font	Matching SAS/GRAPH Font
QCFONT1	SIMPLEX
QCFONT2	DUPLEX
QCFONT3	SWISSE
QCFONT4	SWISS

Choose the special font corresponding to the SAS/GRAPH font in the table that most closely matches the font you are using for general text.

See page 1959 for an example in which QCFONT4 is used to create a title for a control chart.

The following figures illustrate the four special fonts. In each of the figures, the symbols are shown in the special font, and the title and the character codes are shown in the matching SAS/GRAPH font.

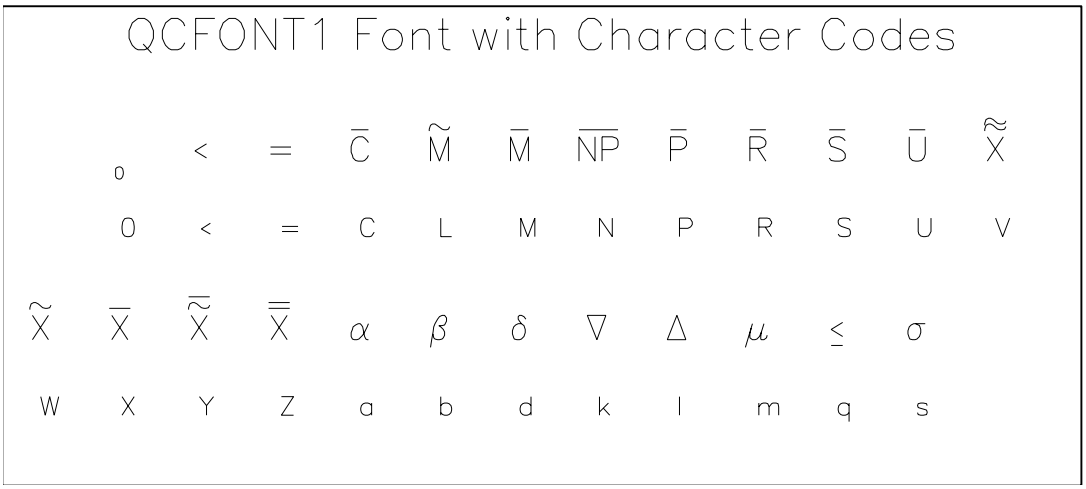


Figure D.1. QCFONT1 and SIMPLEX Fonts

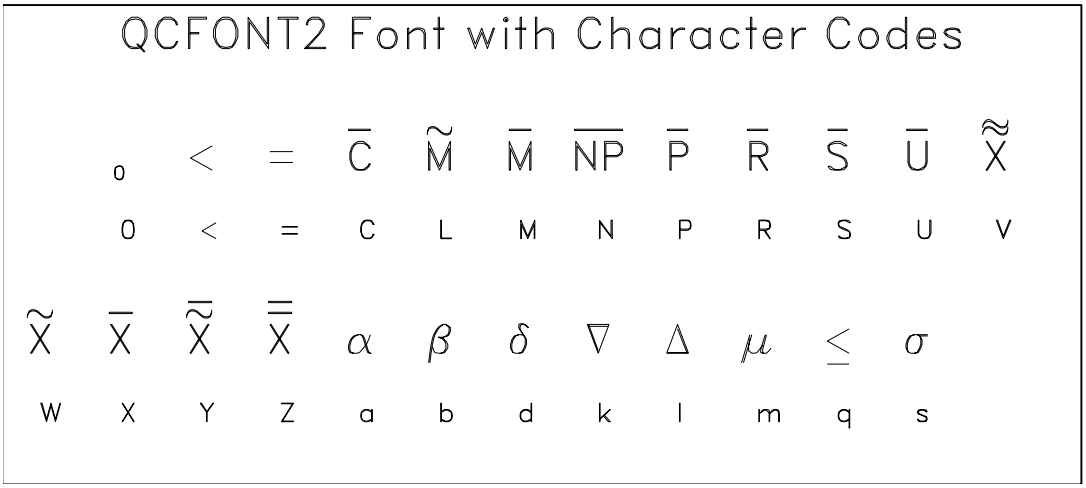


Figure D.2. QCFONT2 and DUPLEX Fonts

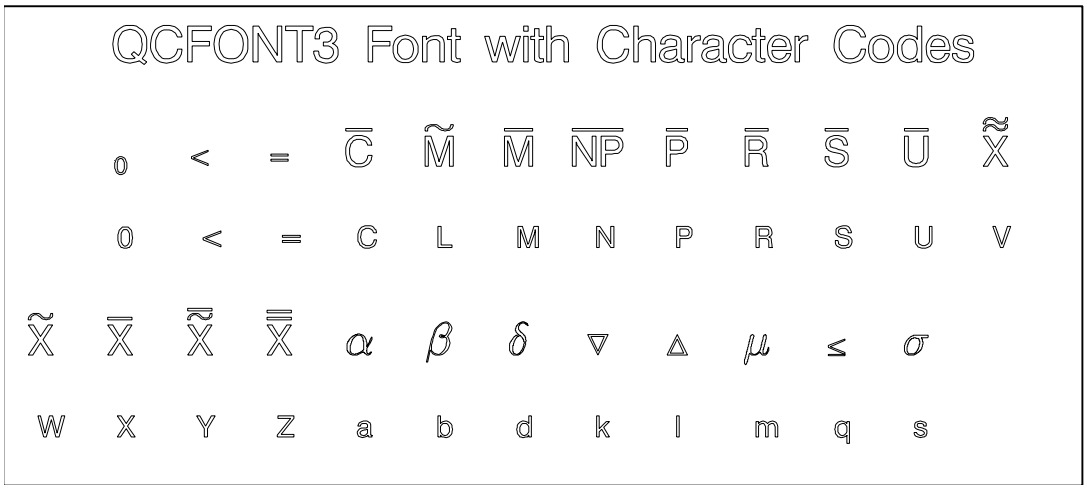


Figure D.3. QCFONT3 and SWISSE Fonts

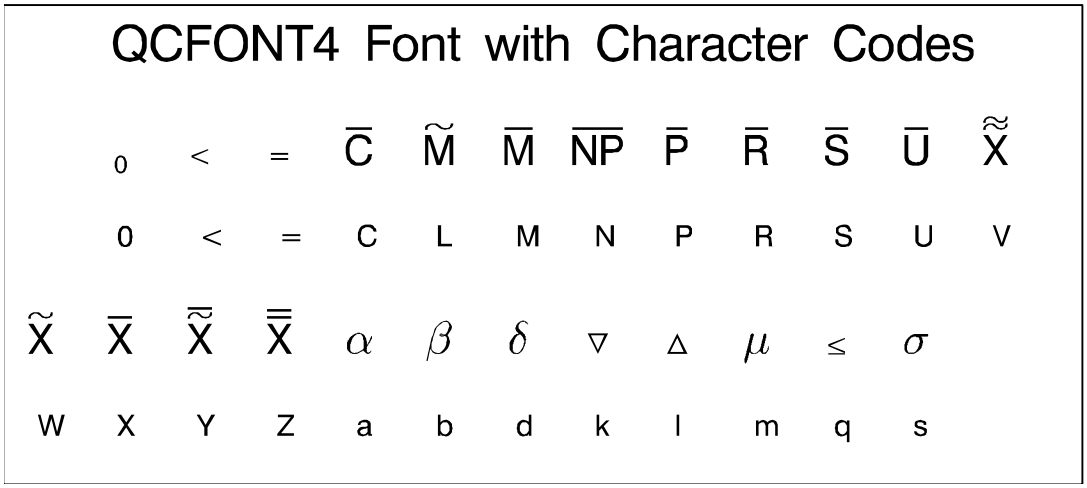


Figure D.4. QCFONT4 and SWISS Fonts

Appendix E
References

American Society for Quality Control (1983), *ASQC Glossary and Tables for Statistical Quality Control*, 230 W. Wells Street, Milwaukee, Wisconsin 53203.

ASQC Automotive Division/AIAG (1990), *Measurement Systems Analysis Reference Manual*, AIAG.

American Society for Testing and Materials (1976), *ASTM Manual on Presentation of Data and Control Chart Analysis*, 1916 Race Street, Philadelphia, PA 19103.

Barrentine, L. (1991), *Concepts for R&R Studies*, Milwaukee, WI: ASQC Quality Press.

Box, G.E.P. and Cox, D.R. (1964), "An Analysis of Transformations," *Journal of the Royal Statistics Society*, B-26, 211–252.

Box, G.E.P., Hunter, W.G., and Hunter, J.S. (1978), *Statistics for Experimenters*, New York: John Wiley & Sons, Inc.

Box, G.E.P. and Meyer, R.D. (1986), "An Analysis for Unreplicated Fractional Factorials," *Technometrics*, 28, 11–18.

Cochran, W.G. and Cox, G.M. (1957), *Experimental Designs, Second Edition,* New York: John Wiley & Sons, Inc.

Cornell, J.A. (1981), *Experiments with Mixtures*, New York: John Wiley & Sons, Inc.

Crowder, S. (1987a), "A Simple Method for Studying Run Length Distributions of Exponentially Weighted Moving Average Charts," *Technometrics,* 29, 401–408.

Crowder, S. (1987b), "Average Run Lengths of Exponentially Weighted Moving Average Charts," *Journal of Quality Technology,* 19, 161–164.

David, H.A. (1981), *Order Statistics*, Second Edition, New York: John Wiley & Sons, Inc.

Duncan, A. (1974), *Quality Control and Industrial Statistics*, Homewood, IL: Richard D. Irwin Inc.

Goel, A. L. and Wu, S. M. (1971), "Determination of A.R.L. and a Contour Nomogram for Cusum Charts to Control Normal Mean," *Technometrics*, 13, 221–230.

Johnson, N.L. and Kotz, S. (1969), *Discrete Distributions*, New York: John Wiley & Sons, Inc.

Kendall, M. and Stuart, A. (1977), *The Advanced Theory of Statistics, Vol. I, Fourth Edition,* New York: Macmillan Publishing Co.

Kume, H. (1985), *Statistical Methods for Quality Improvement*, Tokyo: AOTS Chosakai, Ltd.

Lucas, J. M. and Crosier, R. B. (1982), "Fast Initial Response for CUSUM Quality-Control Schemes: Give Your CUSUM a Head Start," *Technometrics*, 24, 199–205.

McLean, R.A. and Anderson, V.L. (1966), "Extreme Vertices Design of Mixture Experiments," *Technometrics*, 8, 447–454.

Montgomery, D. C. (1996), *Introduction to Statistical Quality Control, Third Edition*, New York: John Wiley & Sons, Inc.

Myers, R.H. (1976), *Response Surface Methodology*, Blacksburg, Virginia: Virginia Polytechnic Institute and State University.

Plackett, R.L. and Burman, J.P. (1947), "The Design of Optimum Multifactorial Experiments," *Biometrika*, 33, 305–325.

SAS Institute Inc. (1999), *SAS/STAT User's Guide, Version 8,* Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1999), *SAS Screen Control Language: Usage, Version 8*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1992), SAS Technical Report P-229, *SAS/STAT Software: Changes and Enhancements, Release 6.07*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1999), *SAS/AF Software: FRAME Entry Usage and Reference, Version 8*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1999), *Getting Started with the ADX Interface for Design of Experiments*, Cary, NC: SAS Institute Inc.

Sarhan, Ahmed H. and Greenberg, Bernard G. (1962), *Contributions to Order Statistics*, New York: John Wiley & Sons, Inc.

SEMATECH, Inc. (1991), *Introduction to Measurement Capability Analysis: Course Notes, Revision 1.2*, Austin, TX: SEMATECH, Inc.

Schilling, E. G. (1982), *Acceptance Sampling in Quality Control*, New York: Marcel Dekker, Inc.

Snee, R.D. (1975), "Experimental Designs for Quadratic Models in Constrained Mixture Spaces," *Technometrics*, 17, 149–159.

Snee, R.D. and Marquardt, D.W. (1974), "Extreme Vertices Designs for Linear Mixture Models," *Technometrics*, 16, 391–408.

Tippett, L. H. C. (1925), "On the Extreme Individuals and the Range of Samples Taken from a Normal Population," *Biometrika*, 17, 364–87.

Wadsworth, H. M., Stephens, K. S., and Godfrey, A. B. (1986), *Modern Methods for Quality Control and Improvement*, New York: John Wiley & Sons, Inc.

Subject Index

A

A-optimal designs,
    See optimal designs, optimality criteria
aberration of a design,
    See minimum aberration
acceptance probability
    double-sampling plan, 2104, 2105
    PROBACC2 function, 2104
    Type A sampling, 2107, 2109, 2110, 2113, 2114
    Type B sampling, 2106, 2107, 2113, 2114
acceptance sampling
    average outgoing quality, 2092, 2093, 2113, 2114
    average sample number, 2093, 2095, 2113, 2115
    average total inspection, 2095, 2096, 2113, 2114
    evaluating double-sampling plans, 2115
    evaluating single-sampling plans, 2113, 2114
    probability of choosing nonconforming items, 2106, 2108, 2109
    types of sampling plans, 2113
alias structure
    breaking links, example, 621–623
    details, 656
    example, 618–620, 635–637
    listing with GLM procedure, 918
    syntax, 608
analysis of variance, 647
Anderson-Darling statistic, 196, 324
Anderson-Darling test, 181
annotating
    cdf plots, 233
    comparative histograms, 257
    example, 861, 862
    histograms, 292
    P-P plots, 415
    probability plots, 441
    Q-Q plots, 471
    Shewhart charts, 1855
ANOM boxcharts
    axis labels, 139
    box-and-whisker plots, description of, 127
    central line, 128
    decision limit equations, 128, 130
    examples, advanced, 139
    examples, introductory, 111
    missing values, 139
    notation, 127
    ODS tables, 133
    options summarized by function, 121–125, 127
    overview, 111
    reading group summary statistics, 114–116, 136, 138
    reading preestablished decision limits, 135, 136
    reading raw measurements, 112–114, 135
    reading summary statistics and decision limits, 119, 120, 138
    saving decision limits, 118, 131
    saving group summary statistics, 117, 118, 131, 132
    saving summary statistics and decision limits, 119, 120, 132, 133
    syntax, 120
ANOM charts for a Two-Way Layout
    central line, 34
    decision limit equations, 34
    notation, 33
    plotted points, 33
ANOM charts for means
    axis labels, 41
    central line, 31
    decision limit equations, 31, 33, 35
    examples, advanced, 41
    examples, introductory, 15
    missing values, 41
    notation, 31
    ODS tables, 37
    options summarized by function, 24–28, 30
    overview, 15
    plotted points, 31
    reading group summary statistics, 18–20, 39, 40
    reading preestablished decision limits, 38, 39
    reading raw measurements, 15, 17, 18, 37, 38
    reading summary statistics and decision limits, 22, 23, 40
    saving decision limits, 21, 35
    saving group summary statistics, 20, 21, 36
    saving summary statistics and decision limits, 22, 23, 36, 37
    syntax, 23
ANOM charts for proportions
    central line, 73
    decision limit equations, 73
    decision limit parameters, 74
    examples, advanced, 81
    getting started, 55
    labeling axes, 80

missing values, 80
notation, 72
ODS tables, 76
options summarized by function, 64–67, 69, 70
overview, 55
plotted points, 72
reading group data, 58, 59, 78, 79
reading group data and decision limits, 63, 79
reading preestablished decision limits, 77, 78
reading raw data, 55, 57, 58, 76, 77
saving decision limits, 61–63, 74, 75
saving group data, 60, 61, 75
saving group data and decision limits, 62, 63, 76
syntax, 63
ANOM charts for rates
central line, 100
decision limit equations, 100, 101
decision limit parameters, 101
examples, introductory, 85
getting started, 85
labeling axes, 106
missing values, 106
notation, 99
ODS tables, 103
options summarized by function, 91, 92, 94, 96, 97
overview, 85
plotted points, 99
reading group data and decision limits, 105, 106
reading number of nonconformities, 105
reading preestablished decision limits, 104
reading raw data, 86, 87, 103
saving decision limits, 88–90, 101
saving group data and decision limits, 102, 103
saving number of nonconformities, 102
syntax, 90
ANOM charts for Rates from Group Counts
examples, advanced, 107
augment, factorial design
example, 618, 621
autocorrelation in process data, 2001–2003, 2005–2009
diagnosing and modeling, 2003, 2005
strategies for handling, 2005–2009
average and range charts,
See $\bar{X}$ and R charts
average and range method
GAGE application, 2067
gage studies, 2059, 2074
average and standard deviation charts,
See $\bar{X}$ and s charts
average chart
GAGE application, 2066
gage studies, 2059, 2073
average charts,
See $\bar{X}$ charts
average outgoing quality
AOQ2 function, 2092
Type B single-sampling, 2113, 2114
average run lengths
cusum schemes, 2099, 2101
EWMA scheme, 2103, 2104
average run lengths (cusum charts),
See cumulative sum control charts
average sample number
ASN2 function, 2093
Type B single-sampling, 2114
average total inspection
ATI2 function, 2095
Type B single-sampling, 2114
axes, Pareto charts, 978, 979, 1011, 1013
axes, Shewhart charts,
See Shewhart charts, axes

B

balanced incomplete block design,
See block designs
balanced lattice, 638
Bayesian optimal designs, 890, 905, 916, 917
beta distribution
cdf plots, 233
chi-square goodness-of-fit test, 323
deviation from empirical distribution, 323
EDF goodness-of-fit test, 323
histograms, 292, 313
histograms, example, 333
P-P plots, 415
probability plots, 441
Q-Q plots, 471
block designs
balanced lattice, examples, 638
optimal designs, examples, 884, 919
randomized complete, examples, 623
block specification, FACTEX procedure
block pseudo-factors, 602, 606
block size, 602
block size restrictions, 607
minimum block size, 602
number of blocks, 602, 607
runs per block, 607
blocking, FACTEX procedure
block pseudo-factor, 665
blocking factor, 665
example, 645
incomplete block design, example, 638
randomization, 652
rename block variable, 613
$\bar{X}$ charts
box appearance, options, 1858, 1861, 1862, 1876, 1886, 1900
box-and-whisker plots, description of, 1266
box-and-whisker plots, style of, 1858
capability indices, computing, 1270
control limit equations, 1267, 1268
control limits, specifying, 1866
displaying points, 1858
examples, advanced, 1284
examples, introductory, 1240

labeling axes, 1283
missing values, 1283
notation, 1266
ODS tables, 1274
options summarized by function, 1254, 1256, 1258, 1259, 1261, 1263
outlier identification color, 1875
outlier identification symbol, 1875
overview, 1239
percentile computation, 1282, 1893
plotting character, 1253
reading preestablished control limits, 1251, 1252, 1275
reading raw measurements, 1240–1243, 1274, 1275
reading subgroup summary statistics, 1243–1246, 1276, 1277
reading summary statistics and control limits, 1250, 1277, 1278
reading summary statistics and decision limits, 133
saving control limits, 1248, 1249, 1269, 1270
saving group summary statistics, 130
saving subgroup summary statistics, 1246–1248, 1270–1272
saving summary statistics and control limits, 1249, 1250, 1273, 1274
schematic box-and-whisker plots, 1288
side-by-side box-and-whisker plots, 1239, 1268, 1287
skeletal box-and-whisker plots, 1287
standard deviation, estimating, 1280–1282
syntax, 1252
tables, creating, 1910

C

c charts
central line, 1328
control limit equations, 1328, 1329
control limit parameters, 1329
examples, advanced, 1337
examples, introductory, 1306
getting started, 1306
known number of nonconformities, specifying, 1340, 1341
labeling axes, 1336
missing values, 1336
notation, 1327
ODS tables, 1332
options summarized by function, 1317, 1318, 1320, 1321, 1324
overview, 1305
plotted points, 1327
plotting character, 1316
reading number of nonconformities, 1311–1313, 1333, 1334
reading preestablished control limits, 1310, 1311, 1333
reading raw data, 1306–1308, 1332
reading subgroup data and control limits, 1334, 1336
saving control limits, 1308, 1309, 1329, 1330
saving nonconformities per unit, 1313, 1315
saving number of nonconformities, 1330
saving subgroup data and control limits, 1331, 1332
syntax, 1315
tests for special causes, 1337, 1338, 1340
candidate data set, OPTEX procedure, See optimal designs, candidate data set
capability indices
$Cp_m(a)$, 182
P_{pk} versus C_{pk}, 203
assumptions, 203
Boyles' index C_{pm}^{+}, 209
computing, 204, 206–208
computing, example, 169
confidence interval, example, 222, 404
confidence limits, 177
estimation from Q-Q plots, 473, 490
estimation from Q-Q plots, example, 499
nonstandard indices, computing, 402
specialized, 208
specification limits, example, 169
specification limits, specifying, 185, 186
terminology, 203
tests for normality, 176
the index C_{jkp}, 209
the index C_{pc}, 214
the index C_{pg}, 213
the index C_{pk}^{W}, 214
the index C_{pm}^{W}, 214
the index C_{pp}, 212
the index $C_{pp}^{''}$, 212
the index C_{pq}, 213
the index C_{p}^{W}, 213
the index k, 208
the index S_{jkp}, 212
the indices $C_{p(5.15)}$, 210
the indices $C_{pk(5.15)}$, 211
the indices C_{pmk}, 211
the indices $C_{pm}(a)$, 210
Vännmann's index $C_p(u, v)$, 215
Vännmann's index $C_p(v)$, 215
Wright's index C_s, 211
CAPABILITY procedure
introduction, 161
learning about, 162
plot statements, 162
cdf plots
annotating, 233
axes, color, 234
axes, specifying, 240
beta distribution, 234
creating, 227
defining character features, 179, 234, 240
example, 227

exponential distribution, 235
font, specifying, 236
gamma distribution, 236
getting started, 227
legends, 237
lognormal distribution, 237
normal distribution, 239
normal distribution, example, 242
options summarized by function, 229–232
overview, 227
reference lines, example, 243
reference lines, options, 235, 237, 238, 240
suppressing empirical cdf, 238
suppressing legend, 238
Weibull distribution, 241
center points, example, 620
chart description, Shewhart charts, 1870
chi-square goodness-of-fit test, 323
compared to EDF test, 342
classification variable,
See comparative histograms
classification variables, OPTEX procedure,
See optimal designs, model
classification variables, Pareto charts, 1051, 1055
clipping points, Shewhart charts,
See Shewhart charts, clipping points
coding designs,
See also optimal designs, coding
coding, FACTEX procedure
block factor, 613
design factor, 612
coefficient of variation
computing, 194
collapsing factors, example, 630
coloring Pareto charts,
See Pareto charts, coloring
coloring, Shewhart charts,
See Shewhart charts, coloring
comparative histograms
annotating, 257
axes, color of, 258
bar labels, specifying, 257, 292
bar width, specifying, 257
bins, specifying, 266
bins, specifying midpoints of, 266
classification variable, missing values of, 266
classification variable, ordering levels of, 268, 269
classification variable, specifying, 259
color, options, 258, 259, 261
columns, number of, 267
font, specifying, 263, 264
getting started, 247
grids, 263
intervals, information about, 270
kernel density estimation, options, 257, 264, 271
legend, 267, 270
line type, grids, 265
normal distribution, example, 249
normal distribution, options, 268, 271
one-way with inset statistics, example, 272
one-way, example, 248
options summarized by function, 253, 254, 256, 257
overview, 247
reference lines, options, 262–265, 271
rows, number of, 268
specification limits, 260
specification limits, filled areas, 185, 186
suppressing plot features, 267, 268
two-way, example, 274
vertical scale, 271
comparative Pareto charts,
See Pareto charts, comparative
computational form of the cusum chart,
See cumulative sum control charts
confidence intervals,
See intervals, CAPABILITY procedure
confidence levels, 176
confidence limits, 176–178
basic parameters, 177
confidence levels, 176
distribution-free, 177
for percentiles, 198
normally distributed, 178
percentiles, 177, 178
probability of exceeding specifications, 178
process capability indices, 177
quantiles, 177, 178
confidence limits, CAPABILITY procedure
confidence level, 177, 178, 183, 1864
type, 177, 178, 182, 183, 1864
confounding rules
compare with alias structure, 656
design factors, 664
details, 656
example, 635
minimum aberration, 657
notation, 656
orthogonally confounded, 666
partial confounding, example, 635
run-indexing factors, 663
searching, 666
syntax, 609
unconfounded effects, 665
connecting points, Shewhart charts, 1866
constants
using functions to calculate, 2115
constants, control charts
A2, 2115
A3, 2115
B3, 2115
B4, 2115
B5, 2115
B6, 2115
c4, 2098
c5, 2115
D1, 2115

D2, 2115
d2, 2101
D3, 2115
d3, 2102
D4, 2115
E2, 2115
E3, 2115
constrained mixture designs,
See mixture designs
contamination, variance
BAYESACT call, 2096
control chart functions
expected value of range, 2101
standard deviation of range, 2102
control factor design, 655
control factors, 655
control factors, example, 642
control limits, Shewhart charts,
See Shewhart charts, control limits
correlated runs, designs with,
See optimal designs, optimal blocking
covariance, optimal designs with,
See optimal designs, optimal blocking
covariates, optimal designs with,
See optimal designs, optimal blocking
Cramér-von Mises statistic, 197
Cramér-von Mises test, 181
Cramer-von Mises statistic, 325
cumulative distribution,
See cdf plots
cumulative percent curve,
See Pareto charts, cumulative percent curve
cumulative sum control charts
annotating, 514
average run length approach, 557–559
central reference value, 558
color, options, 545
compared with Shewhart charts, 560
computational form, 528–531
cusum schemes, specifying, 549
decision interval, defining, 554, 555
designing a cusum scheme, 557–559
detecting shifts, 545, 549
economic design, 558
error probability approach, 558
examples, advanced, 569
examples, introductory, 521
FIR (fast initial response) feature, 553
graphics catalog, specifying, 515
headstart values, 546, 553
interpreting one-sided charts, 555
interpreting two-sided charts, 523, 557
introduction, 509
learning about, 510
line printer features, 514, 515
line types, options, 547
line widths, options, 550
lineprinter plots, using, 516
lower cumulative sum, 552
missing values, 569
monitoring variability, example, 569–571
negative shifts, 552
nonstandardized data, 545
notation, 551
ODS tables, 566
one-sided (decision interval) schemes, 528–531, 552
options summarized by function, 537–540, 542–544
origin, specifying, 547
overview, 521
plotting character, 536
positive shifts, 552
process mean, specifying, 547
process standard deviation, specifying, 549
reading cusum scheme parameters, 516, 533, 534, 567, 568
reading raw measurements, 514, 522, 523, 566, 567
reading subgroup summary statistics, 515, 516, 525, 526, 568, 569
reference values, specifying, 546
saving cusum scheme parameters, 531, 532, 564
saving subgroup summary statistics, 527, 528, 565
saving summary statistics and cusum parameters, 565
Shewhart charts, combined with, 574, 575
standard deviation, estimating, 549, 561–563
suppressing average run length calculation, 547
suppressing display of V-mask, 547
syntax, 514, 535
two-sided (V-mask) schemes, 553, 554
two-sided (V-mask) schemes, examples, 522, 523, 525, 526
Type 1 error probabilities, 544, 549
Type 2 error probabilities, 545
upper and lower cumulative sum charts, combining, 572, 573
upper cumulative sum, 552
V-mask, defining, 555–557
curvature, check for, example, 620
cusum charts,
See cumulative sum control charts
cusum schemes
designing with CUSUMARL function, 2099, 2100

D

D-optimal designs,
See optimal designs, optimality criteria
density estimation,
See kernel density estimation
derived factors, FACTEX procedure
creating, 614
example, 629
descriptive statistics
computing, 192, 193

printing, example, 166
using PROC CAPABILITY, 166
design augmentation, 883, 890, 901, 914
design characteristics, FACTEX procedure
alias structure, 603, 656
confounding rules, 603, 656
design listing, 609
design criteria,
See optimal designs, optimality criteria
design size specification, FACTEX procedure
fraction, 601, 616
minimum runs, 601, 616
number of runs, 601, 616
run indexing factors, 601, 616
syntax, 615
design size specification, OPTEX procedure, 901
design, factorial,
See factorial design
DETMAX algorithm,
See optimal designs, search algorithms
distance from a point to a set, 942
distance-based designs,
See optimal designs, space-filling designs
double-sampling plans,
See acceptance sampling

E

EDF,
See empirical distribution function
effect length, FACTEX procedure
limit, 605
effect length, OPTEX procedure
limit, 892
empirical distribution function
definition of, 196, 323
EDF test compared to chi-square goodness-of-fit test, 342
EDF test statistics, 195, 196, 323
EDF test statistics, Anderson-Darling, 196, 324
EDF test statistics, Cramér-von Mises, 197
EDF test statistics, Cramer-von Mises, 325
EDF test statistics, Kolmogorov-Smirnov, 196, 324
EDF test, probability values, 325
EWMA charts
asymptotic control limits, displaying, 787
asymptotic control limits, example, 807
average run lengths, computing, 815
axis labels, 804
central line, 791
control limit equations, 791
control limits, computing, 786, 791
displaying subgroup means, example, 813
examples, advanced, 805
examples, introductory, 766
missing values, 805
notation, 790
ODS tables, 798
options summarized by function, 778–781, 783–785
overview, 765
plotted points, 790
plotting character, 778
plotting subgroup means, 787
probability limits, 786
process mean, specifying, 787
process standard deviation, specifying, 789
reading preestablished control limit parameters, 775, 776, 799, 800
reading probability limits, 788
reading raw measurements, 766–768, 799
reading subgroup summary statistics, 769–771, 800, 801
reading summary statistics and control limits, 774, 801, 802
saving control limit parameters, 772, 773, 796
saving subgroup summary statistics, 771, 772, 797
saving summary statistics and control limits, 773, 774, 797, 798
specifying parameters for, 805, 806
standard deviation, estimating, 802–804
syntax, 777
varying subgroup sample sizes, 808
weight parameter, choosing, 792
weight parameter, specifying, 789
examine design, FACTEX procedure,
See design characteristics, FACTEX procedure
examples, FACTEX procedure
advanced, 617
alias links breaking, 618
center points, 620
collapsing factors, 630
completely randomized, 617
derived factors, 629
design replication, 625, 627
fold-over design, 621
full factorial, 590
full factorial in blocks, 593
getting started, 590
half-fraction factorial, 595
hyper-Graeco-Latin square, 631
incomplete block design, 638
minimum aberration, 633
mixed-level, 627, 629
partial confounding, 635
point replication, 625, 627
pseudo-factors, 629
randomized complete block design, 623
RCBD, 623
replication, 625, 627
resolution III design, 621
resolution IV, 633
resolution IV, augmented, 618
sequential construction, 635
exchange algorithm,
See optimal designs, search algorithms

expected value
for range of iid normal variables, 2101, 2102
for standard deviation of iid normal sample, 2098, 2099
exponential distribution
cdf plots, 235
chi-square goodness-of-fit test, 323
deviation from empirical distribution, 323
EDF goodness-of-fit test, 323
histograms, 297, 314
P-P plots, 417
probability plots, 444
Q-Q plots, 474
exponentially weighted moving average charts, See EWMA charts
extreme vertex designs, See mixture designs

F

FACTEX procedure
block specification, 606
block specification options, summary, 601
design factor levels, 609
design size options, summary, 601
design size specification, 615
design specification options, summary, 601
examining design characteristics, 608
factor specification options, summary, 601
features, 589
getting started examples, 590
invoking, 605
learning about FACTEX, 590
listing design factors, 609
model specification, 609
model specification options, summary, 601
output, 611
overview, 589
randomization, 614
replication, 613
resolution, 610
statement descriptions, 605
summary of functions, 601
syntax, 601
using interactively, 597
factor specification, FACTEX procedure
factor names, 601
levels, 601
factorial designs, examples, See examples, FACTEX procedure
balanced lattice, 638, 639
efficiency, 611
fractional factorial, minimum aberration, 657
fractional factorial, theory, 663
mixed-level, 614
orthogonal, 627
replicate, 613
resolution, 610
factors, FACTEX procedure
block factor, 648, 665
block pseudo-factor, 649, 656, 665
derived factor, 649
design factor, 648
design factor coding, 612
design factor levels, 609
design factor names, 609
pseudo-factor, 649
run-indexing factor, 649, 656, 663
types, 648
Fedorov algorithm, See optimal designs, search algorithms
filling area underneath density
histograms, 297
FIR (fast initial response) feature, See cumulative sum control charts
fold-over design, example, 621
folded normal distribution, histograms
example, 347
fonts, customizing, 2119–2121
fonts, hardware, 1562, 1780
fonts, Shewhart charts, 1870
fonts, TrueType, 1562, 1780
frequency data, Pareto charts, 966, 967, 1001–1003
frequency tables, 180
full inspection and ASN2 function, 2093
functions
AOQ2, 2092, 2093, 2115
ASN2, 2093, 2095, 2115
ATI2, 2095, 2096, 2115
BAYESACT call, 2096–2098
C4, 2098, 2099, 2115
CUSUMARL, 2099, 2101
D2, 2101, 2102, 2115
D3, 2102, 2103, 2115
EWMAARL, 2103, 2104
for acceptance sampling, 2091
for control chart analysis, 2091
for sampling plans, 2091
PROBACC2, 2104, 2105, 2115
PROBBNML, 2106, 2107, 2113
PROBHYPR, 2107, 2109, 2110, 2113
PROBMED, 2110, 2111
STDMED, 2111, 2112
summary of, 2091

G

G-optimal designs, See optimal designs, optimality criteria
GAGE application, See gage studies
average and range method, 2067
average chart, 2066
data set format, 2080
entering data, 2062, 2064, 2071
gage catalog, 2061
introduction to, 2059
invoking, 2061
missing data, 2068
range chart, 2064

reading data set, 2071
saving data, 2070
variance components method, 2069
gage catalog, 2061
gage repeatability and reproducibility
average and range method, 2076
introduction to, 2059
variance components method, 2079
gage studies,
See GAGE application
average and range method, 2059, 2074
average chart, 2059, 2073
example, 2060
introduction to, 2059
measurement system, 2059, 2060
missing data, 2080
part-to-part variation, average and range method, 2076
part-to-part variation, average chart, 2066, 2073, 2074
part-to-part variation, variance components method, 2079
range chart, 2059, 2072
repeatability, 2059, 2060
repeatability and reproducibility, 2059
repeatability and reproducibility, average and range method, 2076
repeatability and reproducibility, variance components method, 2080
repeatability, average and range method, 2075
repeatability, range chart, 2064, 2072
repeatability, variance components method, 2079
reproducibility, 2059, 2060
reproducibility, average and range method, 2075
reproducibility, average chart, 2066, 2073
reproducibility, variance components method, 2079
terminology, 2059
variance components method, 2059, 2078
gamma distribution
cdf plots, 236
chi-square goodness-of-fit test, 323
deviation from empirical distribution, 323
EDF goodness-of-fit test, 323
histograms, 299, 315
P-P plots, 417, 418
probability plots, 445
Q-Q plots, 474, 475
geometric moving average charts,
See EWMA charts
getting started, ANOM procedure
adding insets to plots, 143
getting started, CAPABILITY procedure
adding insets to plots, 355
creating histograms, 279
cumulative distribution plot, 227
distribution of variable across classes, 247
prediction, confidence, and tolerance intervals, 379
probability plot, 431
probability-probability plot, 410
quantile-quantile plot, 463
saving summary statistics, 393
summary statistics for process capability, 166
getting started, CUSUM procedure
adding insets to plots, 579
getting started, MACONTROL procedure
adding insets to plots, 865
getting started, PARETO procedure
adding insets to plots, 1033
getting started, SHEWHART procedure
adding insets to plots, 1836
Gini's mean difference, 181
GLM procedure, 647, 648
goodness-of-fit test,
See empirical distribution function
See chi-square goodness-of-fit test
Graeco-Latin square, 632
graphical output, Pareto charts, 959
graphics catalog, specifying
CAPABILITY procedure, 180
grid options, Shewhart charts, 1870, 1871, 1877, 1923

H

hanging histograms, 299
HBAR charts
options summarized by function, 1005–1010
syntax, 1005
headstart values in cusum schemes, 2100
histograms,
comparative, See comparative histograms
S_B distribution, 308, 315
S_L distribution, 302
S_N distribution, 306
S_U distribution, 309, 317
adding summary statistics, 284
annotating, 292
axis color, 295
axis scaling, 311
bar width, 304
bars, suppressing, 305
beta distribution, 292, 313
beta distribution, example, 333
capability indices, based on fitted distribution, 301
capability indices, based on fitted distribution, computing, 325, 327
capability indices, based on fitted distribution, example, 343, 344
changing midpoints, example, 284
chi-square goodness-of-fit for fitted distribution, 323
color, options, 295, 296
endpoints of intervals, 308
exponential distribution, 297, 314
filling area underneath density, 297

folded normal distribution, annotating, 347
gamma distribution, 299, 315
getting started, 279
graphical enhancements, 332
interval midpoints, 328
Johnson S_B distribution, 308, 315
Johnson S_L distribution, 302
Johnson S_N distribution, 306
Johnson S_U distribution, 309, 317
kernel density estimation, 319
kernel density estimation, example, 344
kernel density estimation, options, 294, 301, 303, 310, 311
legend, options, 296, 302, 306
legends, suppressing, 305, 306
line type, 302
lognormal distribution, 302, 318
midpoints, 303, 304
multiple distributions, example, 336
normal distribution, 306, 319
normal distribution, example, 280
ODS tables, 331
options summarized by function, 286–289
output data sets, 307, 328, 330, 331
overview, 279
percentile axis, 307
percentiles, 328
plots, suppressing, 306
printed output, 321–325, 327, 328
printed output, capability indices based on fitted distribution, 325, 327
printed output, intervals, 328
printed output, suppressing, 305, 306
quantiles, 307, 328
reference lines, options, 295, 296, 300–303, 311
saving curve parameters, 329
saving goodness-of-fit results, 329
specification limits, color, 185
specification limits, example, 279
specification limits, filled areas, 186
symbols for curves, 310
three-parameter lognormal distribution, example, 345
three-parameter Weibull distribution, example, 347
tick marks on horizontal axis, 300
Weibull distribution, 312, 319
hyper-Graeco-Latin square, example, 631

I

incomplete block design,
See block designs
independent estimate of error, examples, 620, 625
individual measurement and moving range charts
axis labeling, 1381
capability indices, computing, 1373, 1374
central line, 1371
control limit equations, 1372
examples, advanced, 1382
examples, introductory, 1348
interpreting, 1380
missing values, 1381
moving range calculation, controlling, 1356
notation, 1371
ODS tables, 1376
options summarized by function, 1358, 1359, 1361, 1363, 1365, 1366, 1368, 1369
overview, 1347
plotted points, 1371
plotting character, 1358
reading measurements, 1348–1350, 1376
reading measurements and ranges, 1351, 1352, 1377, 1378
reading measurements, ranges, and control limits, 1354, 1378, 1379
reading preestablished control limits, 1354, 1355, 1376, 1377
saving control limits, 1352, 1373, 1374
saving measurements and ranges, 1350, 1374
saving measurements, ranges, and control limits, 1353, 1374, 1375
standard deviation, estimating, 1379, 1380
standard values, specifying, 1383, 1386
syntax, 1357
tests for special causes, 1382, 1383
univariate plots, displaying, 1386, 1387
information matrix, 891, 900
initialization for design search,
See optimal designs, initialization
inner array, 642, 655
input data sets, Shewhart charts,
See Shewhart charts, input data sets
insets
background color, 150, 368, 1041, 1845
background color of header, 151, 369, 1041, 1845
displaying summary statistics, example, 143, 355, 1033, 1836
drop shadow color, 151, 369, 1041, 1845
formatting values, example, 145, 357, 1035, 1838
frame color, 151, 369, 1041, 1845
getting started, 143, 355, 579, 865, 1033, 1836
goodness-of-fit statistics, example, 374
header text color, 151, 369, 1041, 1845
header text, specifying, 146, 151, 358, 369, 1037, 1042, 1839, 1846
labels, example, 145, 357, 1035, 1838
legend, example, 375
overview, 143, 355, 579, 865, 1033, 1835
positioning, details, 152–156, 370–374, 1043–1046, 1847–1850
positioning, example, 146, 358, 1037, 1839
positioning, options, 151, 152, 369, 370, 1042, 1846
statistics associated with distributions, 363–366
summary statistics grouped by function, 149, 362, 363, 1040, 1842

suppressing frame, 152, 370, 1042, 1846
text color, 151, 369, 1042, 1846
interaction, FACTEX procedure
alias structure, 656
between control and noise factors, 645
confounding, 664
examples, 635, 646, 647
generalized, 627, 664, 666
minimum aberration, 657
minimum aberration, example, 633
nonnegligible, 664
resolution, 651
specify terms, 610, 649
interquartile range, 181
saving in output data set, 400
intervals
ODS tables, 390
intervals, CAPABILITY procedure
computing for process capability analysis, 383
computing intervals, example, 379
confidence levels, specifying, 384
confidence, for mean, 385, 388
confidence, for standard deviation, 385, 388
list of options, 384
notation used in computing, 386
number of future observations, 384
one-sided limits, example, 382
prediction, for future observations, 385, 386
prediction, for mean, 385, 387
prediction, for standard deviation, 385, 388
prediction, k-values for, 384
saving information, output data set, 385, 389
specifying method used, 385
specifying type of, 386
suppressing output tables, 385
tolerance, 387
tolerance, for proportion of population, 385
tolerance, p-values for, 385
tolerance, specifying proportion of population, 385
Ishikawa diagrams
adding arrows, 691–694
aligning arrows, 709–715
arrow colors, 729, 731–735
arrow heads, 736
arrow line style, 729, 731–735
arrow line width, 729, 731–735
balancing arrows, 709–715
box color, modifying, 728, 729
box shadow, 737
clipboard graphics, 726, 727
color, arrow, 729, 731–735
color, box, 728, 729
color, palette, 729, 731–735
color, text, 736
context-sensitive operations, 677, 689, 690
data collection, 715, 716
data presentation, 715, 716
deleting arrows, 702–704
detail, decreasing, 716–718
detail, increasing, 716–718
Edit menu, 690
editing existing diagrams, 739, 740
editing labels, 694–697
examples, Integrated Circuit Failures, 746
examples, Photo Development Process, 747
examples, Quality of Air Travel Service, 745
exporting diagrams, 726, 727
File menu, 690
fonts, modifying, 727, 728
Help menu, 691
highlighting arrows, 729, 731–735
history, 675
hotspots, 677, 689, 690
isolating arrows, 720, 721
labeling arrows, 694–697
line palette, 729, 731–735
managing complexity, 716–723
merging diagrams, 721–723
mouse sensitivity, 737
moving arrows, 697–702, 707–715
multiple diagrams, displaying, 721–723, 740
notepads, 715, 716
output, bitmaps, 726, 727
output, graphics, 724, 725
output, SAS data set, 738, 742–744
overview, 675
palettes, colors, 729, 731–735
palettes, fonts, 727, 728
palettes, lines, 729, 731–735
printing, bitmaps, 726, 727
printing, SAS/GRAPH output, 724, 725
resizing arrows, 704–707
SAS data set, input, 739, 740, 742–744
SAS data set, output, 738, 742–744
saving, bitmaps, 726, 727
saving, clipboard graphics, 726, 727
saving, graphics, 724, 725
saving, SAS data set, 738
subsetting arrows, 704–707, 729, 731–735
summary of operations, 689–691
swapping arrows, 707–709
syntax, 744
tagging arrows, 704–707, 729, 731–735
terminology, 677
text entry, 694–697
tutorial, 679, 681–685
undo, 702–704
View menu, 691
zooming arrows, 719, 720, 737

J

Johnson S_B distribution
histograms, 308, 315
Johnson S_L distribution
histograms, 302
Johnson S_N distribution
histograms, 306

Johnson S_U distribution
histograms, 309, 317

K

k-exchange algorithm,
See optimal designs, search algorithms
kernel,
See kernel density estimation
kernel density estimation, 319
adding density curve to histogram, 301
area underneath density curve, 262, 297
bandwidth parameter, specifying, 257, 294
color, 261, 296
density curve, width of, 271, 311
example, 344
filling area under density curve, 263, 297
kernel function, specifying type of, 264, 301
line type for density curve, 265, 302
lower bound, specifying, 303
options used with, 264, 302
upper bound, specifying, 310
kernel function,
See kernel density estimation
Kolmogorov-Smirnov statistic, 196, 324
Kolmogorov-Smirnov test, 181
kurtosis
computing, 193
saving in output data set, 399

L

labeling central line, Shewhart charts,
See Shewhart charts, labeling central line
labeling Shewhart charts,
See Shewhart charts, labeling
line types, Shewhart charts,
See Shewhart charts, line types
location parameter
probability plots, 456
Q-Q plots, 488
lognormal distribution
cdf plots, 237
chi-square goodness-of-fit test, 323
deviation from empirical distribution, 323
EDF goodness-of-fit test, 323
histograms, 302, 318, 345
P-P plots, 419
probability plots, 447
Q-Q plots, 476, 477

M

main effect, 649, 651, 664, 665
main effect, examples, 635–637, 646, 647
maximum value
saving in output data set, 399
mean
saving in output data set, 399
mean and range charts,
See $\bar{X}$ and R charts
mean and standard deviation charts,
See $\bar{X}$ and s charts
mean charts,
See $\bar{X}$ charts
measurement system, gage studies, 2059, 2060
measures of location
mode, 203
median
probability function for, 2110
saving in output data set, 399
standard deviation of, 2111
median absolute deviation about the median, 181
median and R charts
axis labels, 1478
central line, 1464
control limit equations, 1464
examples, advanced, 1474
examples, introductory, 1438
labeling axes, 1473
missing values, 1473
notation, 1463
ODS tables, 1468
options summarized by function, 1450, 1454, 1457–1460
overview, 1437
plotted points, 1464
plotting character, 1450
reading preestablished control limits, 1447, 1448, 1469, 1470
reading raw measurements, 1438, 1439, 1469
reading subgroup summary statistics, 1440–1442, 1444, 1470, 1471
reading summary statistics and control limits, 1446, 1471, 1472
saving control limits, 1445, 1465, 1466
saving subgroup summary statistics, 1444, 1445, 1466, 1467
saving summary statistics and control limits, 1446, 1447, 1467, 1468
standard deviation, estimating, 1472, 1473
syntax, 1449
median and range charts,
See median and R charts
median charts
capability indices, computing, 1420
central line, 1418
control limit equations, 1418
controlling value of central line, 1427
examples, advanced, 1427
examples, introductory, 1392
labeling axes, 1426
missing values, 1426
notation, 1417
ODS tables, 1422
options summarized by function, 1406–1408, 1410, 1411, 1413, 1414
overview, 1391
plotted points, 1417
plotting character, 1406

reading preestablished control limits, 1403, 1404, 1423, 1424
reading raw measurements, 1392, 1393, 1422, 1423
reading subgroup summary statistics, 1394–1396, 1398, 1424, 1425
reading summary statistics and control limits, 1402, 1425, 1426
saving control limits, 1400, 1401, 1419, 1420
saving subgroup summary statistics, 1398–1400, 1420, 1421
saving summary statistics and control limits, 1401, 1402, 1421, 1422
standard deviation, estimating, 1426
syntax, 1405
minimum aberration, 657
aberration vector, 657
blocked design, 658
example, 633
limitation, 634
minimum value
saving in output data set, 399
missing values
CAPABILITY procedure, 215
CUSUM procedure, 569
MACONTROL procedure, 805
output data set, 399
SHEWHART procedure, 1776
mixed-level, factorial design
construction, examples, 627–631
derived factors, 614
mixture designs
examples, 884, 926
plotting, 927–930
mixture-process designs,
See mixture designs
mode
saving in output data set, 399
model specification, FACTEX procedure
directly, 609
estimated effects, 602, 610
indirectly, 609
minimum aberration, 602, 611
nonnegligible effects, 602, 610
resolution, 602, 610
resolution, maximum, 610
specifying effects, 649
modes, 180
modified Fedorov algorithm,
See optimal designs, search algorithms
moving average control charts,
See EWMA charts
See uniformly weighted moving average charts
adding features to, 758
average run lengths, displaying, 861
graphics catalog, specifying, 759
introduction, 753
learning about, 754
line printer features, 758, 759
lineprinter plots, creating, 760
reading control limit parameters, 760
reading raw measurements, 758
reading subgroup summary statistics, 760
syntax, 758
moving range charts,
See individual measurement and moving range charts
multi-vari charts
examples using the SHEWHART procedure, 1298
multivariate control charts, 2033–2037
chart statistic, calculating, 2033
principal component contributions, 2036
mutually orthogonal Latin square, 632, 639

N

neighbor-balanced designs, 926
Newton-Raphson approximation
gamma shape parameter, 233, 292, 303
Weibull shape parameter, 234, 295, 296, 303
noise factors, 642, 655
nonconforming items
probability of choosing, 2106, 2108, 2109
nonnormal process data, 2027, 2028, 2030–2032
calculating probability limits, 2030
preliminary chart, 2028
normal distribution
cdf plots, 239
cdf plots, example, 242
chi-square goodness-of-fit test, 323
comparative histograms, 268
comparative histograms, example, 249
deviation from empirical distribution, 195, 323
EDF goodness-of-fit test, 195, 323
histograms, 305, 306, 319
histograms, example, 280
P-P plots, 420
P-P plots, example, 410
probability plots, 448, 449
Q-Q plots, 478
normal random variables
expected value of standard deviation, 2099
standard deviation of range, 2101
normality tests, 181, 194
Anderson-Darling test, 181
changes made to, 195
Cramér-von Mises test, 181
Kolmogorov-Smirnov test, 181
Shapiro-Wilk test, 181
np charts
central line, 1507
control limit equations, 1507
control limit parameters, 1508
control limits, specifying, 1521–1523
examples, advanced, 1515
getting started, 1484
labeling axes, 1514
missing values, 1514

notation, 1506
ODS tables, 1511
options summarized by function, 1495, 1497, 1499, 1500, 1502, 1503
overview, 1483
plotted points, 1506
plotting character, 1495
reading preestablished control limits, 1492, 1493, 1512, 1521–1523
reading raw data, 1484–1486, 1511
reading subgroup data, 1486–1488, 1512, 1513
reading subgroup data and control limits, 1490, 1492, 1513, 1514
saving control limits, 1489, 1490, 1508
saving subgroup data, 1488, 1489, 1509
saving subgroup data and control limits, 1490, 1509, 1510
standard average proportion, specifying, 1517, 1518
syntax, 1494
tests for special causes, 1515, 1516
unequal subgroup sample sizes, 1518–1520
null hypothesis
location parameter, 180

O

observation exclusion, 178
OC Curve, 1567, 1731
ODS tables
CAPABILITY procedure, 216
FACTEX procedure, 659
OPTTEX procedure, 948
RELIABILITY procedure, 1213, 1214
one-way comparative Pareto charts,
See Pareto charts, comparative
Operating Characteristic Curve, 1567, 1731
optimal blocking,
See optimal designs, optimal blocking
optimal designs
A-efficiency, 936
Bayesian optimal designs, 890, 905, 917
covariate designs, 889, 905
customizing design search, 901
D-efficiency, 936
data set roles, 931, 932
design augmentation, 883, 890, 901, 914
design augmentation data set, 931, 932
design characteristic options, summary, 890
design listing, 891, 900
design search defaults, 901
efficiency measures, 936
efficiency measures, comparing, 906, 907, 909
efficiency measures, interpreting, 937
epsilon value, 892
evaluating an existing design, 903, 923, 944, 946
examining, 900
G-efficiency, 936
getting started examples, 877
including identification variables, 904, 931–933
information matrix, 891, 900
input data sets, 931
interactively, 900, 908
invoking, 892
learning about OPTEX procedure, 876
listing options, summary, 891
memory usage, 942
mixture designs, 926
number of design points, 890, 901, 904
number of search tries, 901, 903
number of tries to keep, 903
OPTEX procedure features, 875
OPTEX procedure overview, 875
optimal blocking, 946
output, 947
output data set, 933
prior precision values, 905, 917
random number seed, 893
resolution IV designs, 917
run-time considerations, 942
saturated design, 882, 904
saving options, summary, 891
search methods, 943
search strategies, 946
statement descriptions, 892
status of search, 893
summary of functions, 890, 891
syntax, 889
treatment candidate points, 923
variance matrix, 891, 900
optimal designs, candidate data set
creating with DATA step, 884, 907
creating with FACTEX procedure, 882, 883
creating with PLAN procedure, 877, 878, 910
discussion, 931
examples of creating, advanced, 906
examples of creating, introductory, 877
recommendations, 915, 946
specifying, 892
optimal designs, coding
default coding, 938
discussion, 937
examples, 938
no coding, 939
orthogonal coding, 890, 921, 922, 924, 938
recommendations, 939
specifying, 892
static coding, 890, 938
summary of options, 890
optimal designs, examples
advanced, 906
Bayesian optimal designs, 916
block design, 884, 919
design augmentation, 883, 914
designs with correlated runs, 924
designs with covariates, 921
handling many variables, 883
initialization, 912
introductory, 877

mixture design, 884, 926
nonstandard modeling, 906
reducing candidate set, 915
resolution IV design, 916
saturated second-order design, 882
using different search methods, 910
optimal designs, initialization
defaults, 901–903
example, 912
initial design data set, 902, 912, 931, 932
optimal blocking, 894
partially random, 902
random, 902
recommendations, 947
sequential, 902
specifying, 902
summary of options, 890, 891
optimal designs, model
abbreviation operators, 935
classification variables, 895, 934
crossed effects, 935
discussion, 933
examples, 936
factorial model, 936
interactions, 935
main effects, 934
main effects model, 936
no-intercept model, 890, 905
nonstandard, 906
polynomial effects, 934
quadratic model, 936
regressor effects, 934
specifying, 890, 904
summary of options, 890
types of effects, 904, 934
types of variables, 933
optimal designs, optimal blocking
A-efficiency, 937
block specification, 894
classification variables, 895
covariance specification, 894
covariate designs, 921
D-efficiency, 937
data sets, 933
discussion, 946
evaluating an existing design, 946
examples, 919, 921, 924
initialization, 894
number of search tries, 894
specifying, 890, 893
summary of options, 890
suppressing exchange step, 894
treatment candidate points, 893, 923
tries to keep, 894
optimal designs, optimality criteria
A-optimality, 901, 909, 940
computational limitations, 942
D-optimality, 901, 940
default, 901
definitions, 940–942
discussion, 939
distance-based, 940, 942
examples, 907, 926
G-optimality, 906, 941
I-optimality, 941
information-based, 939
S-optimality, 902, 942
specifying, 890, 901, 902
summary of options, 890
types, 939
U-optimality, 901, 926, 942
optimal designs, output
block variable name, 891, 906
design number, 906
options, 906
output data set, 905, 933
selecting design by efficiency, 906, 941
transfer variables, 891, 904
optimal designs, search algorithms
comparing different algorithms, 910, 911
default, 901
DETMAX, 903, 910, 911, 945
discussion, 943
example, 910, 911
exchange, 903, 945
excursion level for DETMAX, 903
Fedorov, 903, 945
k-exchange, 903, 945
modified Fedorov, 903, 945
rank-one updates, 943
sequential, 903, 910, 911, 944
specifying, 891, 903
speed, 904, 910, 911, 943
summary, 891
optimal designs, space-filling designs
coding for, 939
criteria, 940
definitions, 942
distance from a point to a set, 942
efficiency measures, 937
examples, 926
S-optimality, 942
specifying, 901, 902
U-optimality, 942
optimality criteria,
See optimal designs, optimality criteria
options, Shewhart charts
dictionary, 1853
orthogonal confounding, 648, 649
orthogonal design
theory, 663
outer array, 642, 655
outgoing quality,
See AOQ2 function
output data set, Pareto charts, 1053, 1054
output data sets, CAPABILITY procedure
creating, 401
getting started, 393

naming, 396
percentile variable names, 397, 398
percentiles, 396
saving capability indices and related statistics, 399
saving specification limits and related statistics, 399
saving summary statistics, 399, 400
saving test statistics, 400
output data sets, Shewhart charts, See Shewhart charts, output data sets
output, FACTEX procedure
code design factor levels, 603, 612
decode block factor levels, 603, 613
decode design factor levels, 603, 612
details, 658
options, 612
output data set, 611, 658
rename block variable, 603, 613
output, OPTEX procedure, See optimal designs, output

P

p charts
central line, 1550
control limit equations, 1550
control limit parameters, 1551
control limits, revising, 1565–1567
examples, advanced, 1558
getting started, 1528
labeling axes, 1557
missing values, 1558
notation, 1549
OC curves, 1567–1569
ODS tables, 1554
options summarized by function, 1539–1541, 1543–1545, 1547
overview, 1527
plotted points, 1549
plotting character, 1538
reading preestablished control limits, 1536, 1555
reading raw data, 1528–1530, 1554
reading subgroup data, 1530–1532, 1555, 1556
reading subgroup data and control limits, 1535, 1556, 1557
saving control limits, 1534, 1535, 1551
saving subgroup data, 1533, 1552
saving subgroup data and control limits, 1535, 1553
standard average proportion, specifying, 1560, 1562
syntax, 1537
tests for special causes, 1559
unequal subgroup sample sizes, 1562–1564
P-P plots
annotating, 415
axes, color of, 416
axes, horizontal, 418
axes, vertical, 421, 422
beta distribution, 415
compared to Q-Q plots, 425
distribution options, 412, 413, 426
distribution reference line, 411, 413
exponential distribution, 417
frame, color of, 416
gamma distribution, 417, 418
general plot layout, 414
getting started, 410
graphics device, options, 414, 427
interpreting, 423
line printer, options, 414, 421
line width, distribution reference line, 422, 427
lognormal distribution, 419
normal distribution, 420
normal distribution, example, 410
options summarized by function, 412–414
overview, 409
reference lines, options, 416, 418, 419, 422
text, color of, 416
Weibull distribution, 422
Pareto charts
avoiding clutter, 1054
axes, 978, 979, 994, 1011, 1013, 1018
axis options, 975, 1009
bars, displaying, 976, 1010
before-and-after, 1056–1059
classification variables, 1051, 1055
dictionary of options, 959
examples, advanced, 1056
examples, introductory, 963, 999
graphics catalog, 959
grids, 974, 984, 1008, 1018
highlighting, 1066–1070
labeling chart features, 1052
large data sets, 1055
levels, 1049
merging columns, example, 1073
missing values, 988, 989, 1022, 1023, 1055
options summarized by function, 958
output data set, 1053, 1054
overview, 953
Pareto curve, 965, 1000
Pareto, Vilfredo, 953
process variables, 964, 1000, 1049, 1055
reading frequency data, 966, 967, 1001–1003
reading raw data, 963–965, 999–1001
reference lines, 973, 1007
restricting number of categories, 968, 970, 972, 1003, 1005, 1006
saving information, 1053, 1054
scaling bars, 993, 1027, 1054
seven basic QC tools, 953
side-by-side, 953
stacked, 953
syntax, 958
tied categories, 968, 970, 1003, 1005
“trivial many”, 953, 1066
“useful many”, 953, 1066

using raw data, example, 963–965, 999–1001
vertical axis, 1049
visual clarity, 1054
"vital few", 953, 1066
Pareto charts, categories, 965, 1000, 1049
legend, 965, 1001
maximum number of, 1055
restricting, 968, 970, 1003, 1005
restricting number of, 987, 988, 1021, 1022
ties, 968, 970, 1003, 1005
unbalanced, 1051
Pareto charts, classification variables
examples, 1056, 1060
Pareto charts, coloring
axes, 979, 1013
bar outlines, 980, 1013
bars, 980, 1014
cumulative percent curve, 980, 1014
grid lines, 981, 1015
highest bars, 981, 1015
labels, 981, 1014
lowest bars, 983, 1016
other category, 983, 1016
recommendations, 1054
reference lines, 981, 984, 1015, 1017
secondary axis, 980, 1013
tick marks, 979, 1013
tiles, 984, 1017
Pareto charts, comparative, 953, 974, 1008, 1050
cells, 1050
classification variables, 1058
classification variables, examples, 1056, 1060
creating, 981, 1015
frequency proportion bars, 983, 1017
key cell, 982, 1016, 1051, 1058, 1066
merging columns, 1073
one-way, 1051
one-way, example, 1064
ordering values, 990, 991, 1024, 1025
rows and columns, ordering, 990, 991, 1024, 1025
tiles, 1051, 1069
two-way, 1051
two-way, examples, 1060, 1065–1067, 1069, 1071, 1073
unbalanced categories, 991, 1025, 1051
weighted charts, 1075
Pareto charts, cumulative percent curve, 965, 990, 1000, 1024, 1049
anchoring, 1061, 1062
coloring, 980, 1014
enhancing, 971, 1006
scaling, 1052
suppressing, 1054, 1063, 1064
Pareto charts, grid lines
width, 995, 1028, 1029
Pareto charts, legends
bar legend labels, 979, 1013
bar legends, 978, 1012
category legend labels, 979, 1013
highest and lowest bars legend labels, 985, 1019
sample size legend color, 981, 1014
sample size legends, 975, 989, 1009, 1023
tile legend labels, 994, 1028
tile legends, 993, 1027
Pareto charts, *other* category, 968, 970, 991, 992, 1003, 1005, 1025, 1026
coloring, 983, 1016
labeling, 986, 1020
pattern, 993, 1027
Pareto charts, restricted, 968, 970, 987, 988, 1003, 1005, 1021, 1022, 1050, 1055
large data sets, 1055
Pareto charts, weighted, 1050
example, 1075
Pareto curve, 965, 1000
Pareto principle, 953
Pareto, Vilfredo, 953
partial confounding, example, 635
pattern tests,
See Shewhart charts, tests for special causes
percent plots,
See P-P plots
percentiles
axes, Q-Q plots, 479, 480, 489
confidence limits, 198
defining, 181, 197
empirical distribution function, 197
saving in output data set, 396
visual estimates, Q-Q plots, 489
weighted, 198
weighted average, 197
PLAN procedure, 640
plot statements, CAPABILITY procedure, 162
prediction intervals,
See intervals, CAPABILITY procedure
probability functions
binomial, 2106, 2107
for median, 2110, 2111
hypergeometric, 2107, 2109, 2110
probability limits, Shewhart charts, 1855, 1895, 1900, 1901
probability of exceeding specifications, 178
probability plots
axes, color, 443
axes, horizontal, 446
axes, rotating, 450
axes, vertical, 452
beta distribution, 441, 442
distribution reference lines, 451, 456
distribution reference lines, examples, 457, 459, 460
distributions, 454
exponential distribution, 444
frame, color, 443
gamma distribution, 445
general plot layout, 440
getting started, 431

graphics device, options, 441
graphics, options, 456
legends, 446
legends, suppressing, 448
line printer, options, 440, 449, 451
location parameter, 456
lognormal distribution, 447
lognormal distribution, example, 434
normal distribution, 437, 448, 449
normal distribution, example, 432
options summarized by function, 438–440
overview, 431
percentile axis, 449
reference lines, 443, 444, 446, 448, 452
scale parameter, 456
shape parameter, 450, 455
syntax, 437
text, color, 444
threshold parameter, 451, 456
Weibull distribution, 452–454
probability-probability plots,
See P-P plots
PROC CAPABILITY statement, 165
process capability indices
confidence limits, 177
process distribution,
See empirical distribution function
process potential
P_{pk} versus C_{pk}, 203
process variables, Pareto charts, 964, 1000, 1049, 1055
pseudo-factors, example, 629

Q

Q-Q plots
axes, color, 473
axes, horizontal, 475
axes, options, 470
axes, percentile scale, 479, 480, 489
axes, rotating, 481
axes, vertical, 482
beta distribution, 469, 471
capability indices, 473, 478, 490, 499
creating, 485
diagnostics, 486
distribution reference lines, 466, 488
distributions, 468, 487
estimating C_{pk}, 499
exponential distribution, 469, 474
frame, color, 473
gamma distribution, 469, 474
general plot layout, 470
getting started, 463
graphics device, options, 471, 490
interpretation, 486
legends, 476
legends, suppressing, 466, 477–479
line printer, options, 470, 480, 482
line width, 490
location parameter, 488
lognormal distribution, 469, 476, 477
lognormal distribution, example, 492
nonnormal data, example, 491
normal distribution, 469, 478
normal distribution, example, 464, 499
options summarized by function, 468, 470, 471
overview, 463
percentiles, estimates, 489
reference lines, 468, 473, 474, 476, 477, 479, 483, 490
sample estimates, 478
scale parameter, 488
syntax, 467
text, color, 473
threshold parameter, 488
Weibull distribution, 469, 483–485
Weibull distribution, example, 496
quantile-quantile plots,
See Q-Q plots
quantiles
defining, 197
empirical distribution function, 197
weighted average, 197

R

R charts
capability indices, computing, 1598
central line, 1595
control limit equations, 1595, 1596
control limits, specifying, 1607, 1608
examples, advanced, 1605
examples, introductory, 1574
labeling axes, 1604
missing values, 1604
notation, 1595
ODS tables, 1600
options summarized by function, 1586–1590, 1592, 1593
overview, 1573
plotted points, 1595
plotting character, 1586
probability limits, 1605, 1606
reading preestablished control limits, 1583, 1584, 1601
reading raw measurements, 1574–1576, 1600
reading subgroup summary statistics, 1576–1579, 1601, 1602
reading summary statistics and control limits, 1582, 1602, 1603
saving control limits, 1580, 1581, 1596, 1598
saving subgroup summary statistics, 1579, 1580, 1598
saving summary statistics and control limits, 1581, 1582, 1599
standard deviation, estimating, 1603, 1604
syntax, 1585
randomization, FACTEX procedure
blocking, 652

details, 652
example, 617, 623
prevent, 614, 653
seed, 614, 623
randomized complete block, example, 623
randomized treatments, example, 623
range
saving in output data set, 399
range chart
GAGE application, 2064
gage studies, 2059, 2072
range charts,
See R charts
reference lines, Shewhart charts,
See Shewhart charts, reference lines
reliability analysis
analyzing accelerated life test data, 1090, 1092–1096
analyzing arbitrarily censored data, 1100
analyzing binomial data, 1129, 1131
analyzing combined failure modes, 1115
analyzing groups of data, 1088–1090
analyzing interval-censored data, 1096–1098, 1102
analyzing regression models, 1104–1107, 1109
analyzing repair data, 1118, 1120, 1122
analyzing right-censored data, 1085, 1086, 1088
analyzing two groups of repair data, 1122–1125
arbitrarily censored data, 1182
binomial parameter estimation, 1197, 1198
classification variables, 1138
confidence intervals for parameters, 1193
covariance matrix of parameters, 1192
creating life-stress relation plots, 1164, 1166, 1167, 1169, 1171–1173
creating output data sets, 1144, 1214
creating probability plots, 1156, 1158, 1160–1162, 1164
details, 1174
estimating distribution parameters, 1134, 1135, 1138
examples, 1085
failure modes, 1115, 1139, 1202
fitting regression models, 1152–1155
frequency variables, 1140
insets, 1141–1143
least squares estimation, 1201
log-scale regression model parameters, 1191
maximum likelihood estimation, 1188
mean cumulative function plots, 1145, 1146, 1148, 1149, 1152, 1173
observation-wise percentiles, 1203–1205
observation-wise predicted values, 1203
observation-wise reliability function estimates, 1206
observation-wise statistics, 1203–1206, 1208
overview, 1083, 1084
parameter estimation, 1188, 1190–1193, 1195–1202
percentile estimation, 1195, 1196
Poisson parameter estimation, 1199, 1200
probability distributions, 1174–1176
probability plots, 1177–1181
readout data, 1156
recurrence data, 1208, 1209, 1213
regression model parameters, 1190, 1191
reliability function estimation, 1196, 1197
residuals, 1206, 1208
specifying failure modes, 1139
specifying probability distributions, 1138
syntax, 1132
Turnbull algorithm, 1182
types of lifetime data, 1174
Weibayes estimation, 1201
repeatability
average and range method, 2075
definition of, 2060
introduction to, 2059
range chart, 2064, 2072
variance components method, 2079
repeatability and reproducibility
average and range method, 2076
introduction to, 2059
variance components method, 2080
replication, FACTEX procedure
data set, 602, 613, 614
design point, 614
design replication, 654, 655
details, 654
entire design, 613
example, 625, 627
fixed number of times, 602, 654
inner array, 655
number of times, 613, 614
outer array, 655
point replication, 654, 655
reproducibility
average and range method, 2075
average chart, 2066, 2073
definition of, 2060
introduction to, 2059
variance components method, 2080
resolution, FACTEX procedure
comparison, 651
definition, 651
example, 595, 618, 633
minimum aberration, 657
number, 651
numbering scheme, 651
syntax, 610
response, factorial design, 647, 648
restricted Pareto charts,
See Pareto charts, restricted
robust estimators
location, 200
scale, 200
trimmed means, 200
Winsorized means, 200

robust measures of scale, 181
Q_n, 181
S_n, 181
rounding, 181
rules for lack of control,
See Shewhart charts, tests for special causes
runs rules,
See Shewhart charts, tests for special causes
runs tests,
See Shewhart charts, tests for special causes

S

s charts
central line, 1634
control limit equations, 1635
examples, advanced, 1644
examples, introductory, 1614
notation, 1634
ODS tables, 1639
options summarized by function, 1625–1630, 1632, 1634
overview, 1613
plotted points, 1634
plotting character, 1625
reading preestablished control limits, 1623, 1624, 1640
reading raw measurements, 1614–1616, 1639
reading subgroup summary statistics, 1617–1619, 1640, 1641
reading summary statistics and control limits, 1622, 1641, 1642
saving control limits, 1620, 1621, 1635, 1637
saving subgroup summary statistics, 1619, 1620, 1637
saving summary statistics and control limits, 1621, 1622, 1638
standard deviation, estimating, 1642, 1643
standard deviation, specifying, 1644, 1645
syntax, 1624
s charts
capability indices, computing, 1637
labeling axes, 1644
missing values, 1644
S-optimal designs,
See optimal designs, space-filling designs
sampling plans,
See also acceptance sampling
double, 2115
single, 2113, 2114
types of, 2113
saturated designs, analysis of, 2097
saturated designs, OPTEX procedure, 882, 904
save design, FACTEX procedure,
See output, FACTEX procedure
S_B distribution
histograms, 308, 315
scale parameter
probability plots, 456
Q-Q plots, 481, 488
search algorithms, optimal designs,
See optimal designs, search algorithms
search design, FACTEX procedure
confounding rules, 666
limit, 602, 606
maximum time, 602, 606
speeding, 667
semicurtailed inspection and ASN2 function, 2093
sequential algorithm,
See optimal designs, search algorithms
seven basic QC tools, 953
shape parameter
probability plots, 450, 455
Q-Q plots, 481, 488
Shapiro-Wilk test, 181
Shewhart charts
annotating, 1855
average run lengths, example, 1646
between-subgroup variance, 2013
capability indices, computing, 1774, 1775
challenging assumptions of, 2001
chart description, 1870
chart naming, 1882
computing capability indices, 1879, 1911, 1919
connecting points, 1866, 1883
control chart statistics, 1881
displaying points, 1854
estimating μ, 1881
estimating σ, 1902, 1908
exceptions charts, 1870, 1910
fonts, 1870
fonts, hardware, 1562, 1780
fonts, TrueType, 1562, 1780
grids, 1870, 1871, 1877, 1923
horizontal axes, 1887
identifying unequal subgroup sample sizes, 1883
intervals between subgroups, 1875
missing values, 1776
options dictionary, 1853
plot margins, 1880, 1899
probability limits, 1855, 1895, 1900, 1901
separating, 1900
separating subgroups, 1893
subgroup sample size, 1909
subgroup-variables, 1771, 1772
subgroups, 1882
vertical axes, 1923
Shewhart charts, axes
appearance, 1923
coloring, 1861
for multiple pages, 1899
horizontal, 1871, 1920
labeling, 1478, 1480, 1966, 1967, 1969
offset length, 1872
scaling on p charts, 1925
scaling primary and secondary charts, 1925
suppressing labels, 1884
tick mark labels, 1901, 1918
tick marks, 1872, 1920, 1921

vertical axis truncation, 1887
Shewhart charts, box charts,
See box charts
Shewhart charts, clipping points, 1863–1865, 1879
examples, 1962, 1963, 1965
Shewhart charts, coloring
axes, 1861
axis labels, 1869
connecting lines, 1863, 1865, 1866
control limits, 1864
frames, 1863
HREF= lines, 1864
inside control limits, 1864
inside stars, 1867
label frames, 1863
outside control limits, 1866
phase labels, 1867
star outlines, 1867, 1868
STARCIRCLES= circles, 1867
TESTS= option, 1869, 1870
tick marks, 1869
VREF= lines, 1869
Shewhart charts, control limits
appearance, 1923
computing, 1855, 1885, 1900
for autocorrelated data, 2001–2003, 2005–2009
for data with multiple components of variation, 2009, 2010, 2012–2016
for nonnormal processes, 2027, 2030–2032
for short-run processes, 2016, 2018, 2019, 2021–2027
labeling, 1877, 1919
line type, 1879
multiple sets, 1939, 1941–1947
observations used in computation, 1972
sample size, 1877, 1878
Shewhart charts, fonts
customizing, 2119–2121
Shewhart charts, for autocorrelated data,
See autocorrelation in process data
Shewhart charts, for data with multiple components of variation,
See variation, multiple components of
Shewhart charts, for multivariate data,
See multivariate control charts
Shewhart charts, for nonnormal process data,
See nonnormal process data
Shewhart charts, for short-run processes,
See short run process control
Shewhart charts, input data sets
control limits, 1895, 1896
probability limits, 1895
specifying blocks, 1897
Shewhart charts, labeling
angles for, 1876
axes, 1478, 1480, 1966, 1967, 1969
control limits, 1877, 1883, 1919
fonts for, 1876, 1912
height for, 1871, 1876, 1913
horizontal axis, 1967, 1969
points, 1854, 1918
points outside control limits, 1889, 1890
reference lines, 1873, 1922
splitting labels, 1902
stars, 1905
tests for special causes, 1913, 1914
tick marks, 1901, 1918
vertical axis, 1888, 1967, 1969
zone lines, 1925, 1926
Shewhart charts, labeling central line
c chart, 1868
m chart, 1924
np chart, 1888
p chart, 1894
r chart, 1899
s chart, 1902
u chart, 1919
x chart, 1924
decimal digits, number of, 1883
Shewhart charts, line types
reference lines, 1881
star outlines, 1880
STARCIRCLES= circles, 1879
TESTS= option, 1880
Shewhart charts, nonnormal process data
example, 1828–1831
Shewhart charts, output data sets
chart information, 1891
control limits, 1889, 1890
indicating parameters as estimates or standard values, 1918
subgroup summary statistics, 1889, 1890
Shewhart charts, pages
maximum, 1881
numbering, 1893
splitting, 1856
Shewhart charts, phase variables
control limits, 1894
delineating, 1894
labels, 1893
legends, 1894
Shewhart charts, reference lines
applying to all BY groups, 1883
horizontal axis, 1872, 1873
label position, 1873, 1922
labels, 1873, 1922
line type, 1877, 1881
symbol, 1873, 1922
vertical axis, 1921, 1922
Shewhart charts, specifying parameters
μ_0, 1882
σ_0, 1900
p_0, 1892
u_0, 1919
Shewhart charts, star charts, 1948–1957
contrasted with multivariate control charts, 1949
Shewhart charts, stars
circle outline width, 1924

creating, 1908
inner radius, 1904
labeling, 1905
legends, 1905, 1906
outer radius, 1903, 1906
process variables, 1948
reference circles, 1903, 1950, 1951
standardizing, 1906, 1907, 1955–1957
star outline width, 1924
style, 1908, 1952–1954
vertex angle, 1907
vertex variables, 1948, 1949
Shewhart charts, stratification of data, 1929–1934, 1936–1938
by a _PHASE_ *variable*, 1936
by a *_PHASE_ variable*, 1937, 1938
by a *symbol-variable*, 1931, 1932
by *block-variables*, 1932–1934, 1936
Shewhart charts, subgroup selection
using switch variables, 1973, 1974
using WHERE statement, 1970–1973
Shewhart charts, suppressing features of
central lines, 1884
connecting line segments, 1884
control limit frames, 1885
control limit legends, 1885
control limits, 1885
entire chart, 1884
frames, 1884
horizontal axis labels, 1884
labels, 1885
legends, 1885
line segments, 1887
lower control limits, 1884, 1885
phase legend frames, 1885
upper control limits, 1885, 1887
Shewhart charts, tables, 1910
adding central line values, 1910
adding control limit exceedances, 1911
adding ID variables, 1911
adding legends, 1911
adding TESTS= results, 1911
box charts, 1910
Shewhart charts, tests for special causes, 1912–1917, 1926
across phases, 1912
customizing tests, 1995–1997
definitions, 1978
generalized patterns, 1992–1995
label angles, 1876
label fonts, 1876, 1912
label height, 1876, 1913
labeling signaled points, 1985, 1990
labels, 1913, 1914
line segment character, 1912
M-patterns, 1992–1995
multiple phases, 1986
multiple sets of control limits, 1987, 1989, 1990
nonstandard tests, 1991–1997
overlapping points, 1914
range and standard deviation charts, 1991, 1992
reset, 1911, 1915
run lengths, 1912
standard tests, 1977–1979, 1981–1987, 1989, 1990
standard tests, interpreting, 1981
standard tests, modifying, 1982
standard tests, requesting, 1979, 1981
suppressing 3-sigma check, 1883
T-patterns, 1992–1995
varying subgroup sample sizes, 1914, 1983, 1984
zone line labels, 1925, 1926
zone lines, 1926
zones, 1926
Shewhart charts, trends
displaying, 1924, 1957, 1959–1961
modeling, 1960, 1961
recognizing, 1959
trend variables, 1917
Shewhart charts, warning limits
vertical axis, 1921
short run process control, 2016, 2018, 2019, 2021–2027
testing for constant variances, 2025
difference from nominal approach, 2018, 2019, 2021–2024
standardization approach, 2026, 2027
side-by-side Pareto charts, 953
sign test, 180
signal-to-noise ratio, 642
signed rank statistic, computing, 194
signed rank test, 180
single-sampling plans,
See acceptance sampling
size specification, FACTEX procedure,
See design size specification
skewness
saving in output data set, 399
S_L distribution
histograms, 302
smoothing data distribution,
See kernel density estimation
S_N distribution
histograms, 306
space-filling designs,
See optimal designs, space-filling designs
specialized capability indices, 182
specification limits, 182
capability indices, confidence interval, 222
comparative histograms, 260
computing capability indices, example, 169
examples, 218
exceeding, 399
histograms, example, 279
identifying, 188
lower limit, specification of, 185
reading from data set, example, 218

reference lines, color of, 185
reference lines, example, 220
reference lines, filled areas, 186
reference lines, line type, 185
reference lines, width of, 187
summary information, 169
suppressing legend for, 239, 306
target line, color of, 185
target line, line type, 186
target value, specification of, 186
upper limit, specification of, 186
stacked Pareto charts, 953
standard deviation
boxcharts, 1895
CAPABILITY procedure, 183
for median of standard normal, 2111, 2112
range of iid normal variables, 2102
saving in output data set, 399
specifying, 239
standard deviation charts,
See s charts
star charts,
See Shewhart charts, star charts
S_U distribution
histograms, 309, 317
subgroup variables
character, 1772
numeric, 1772
dates or *times*, 1771
indices, 1771
sum
saving in output data set, 399
sum of weights
saving in output data set, 399
summary statistics, 176
printing, example, 166
saving, 181
tables, 176
supplementary rules,
See Shewhart charts, tests for special causes
suppressing features of Shewhart charts,
See Shewhart charts, suppressing features of
suspended histograms, 299

T

tables
modes, 180
sign test, 180
signed rank test, 180
trimmed means, 182
Winsorized means, 183
tables, CAPABILITY procedure
summary statistics, 176
tables, Shewhart charts,
See Shewhart charts, tables
test statistics
saving in output data set, 400
tests for normality, 176
tests for special causes, Shewhart charts,
See Shewhart charts, tests for special causes
tests of location
location parameter, 180
threshold parameter
probability plots, 451, 456
Q-Q plots, 482, 488
tolerance intervals,
See intervals, CAPABILITY procedure
trimmed means, 182, 200
two-way comparative Pareto charts,
See Pareto charts, comparative
Type A sampling, 2113
Type B sampling, 2113
Type I sum of squares, 647

U

u charts
central line, 1674
compared with c charts, 1674
control limit equations, 1674, 1675
control limit parameters, 1675
examples, advanced, 1682
examples, introductory, 1652
getting started, 1652
known number of nonconformities, specifying, 1684, 1685
labeling axes, 1681
missing values, 1681
notation, 1673
ODS tables, 1678
options summarized by function, 1664, 1666, 1668–1670, 1672
overview, 1651
plotted points, 1673
plotting character, 1663
reading number of nonconformities, 1657–1659, 1679, 1680
reading preestablished control limits, 1656, 1657, 1679
reading raw data, 1652–1654, 1678, 1679
reading subgroup data and control limits, 1680, 1681
saving control limits, 1654–1656, 1675, 1676
saving nonconformities per unit, 1660, 1661
saving number of nonconformities, 1676, 1677
saving subgroup data and control limits, 1677, 1678
syntax, 1662
tests for special causes, 1682, 1683
unequal subgroup sample sizes, 1685–1688
U-optimal designs,
See optimal designs, space-filling designs
uniformly weighted moving average charts
adding features to, 861, 862
annotating charts, 861, 862
asymptotic control limits, displaying, 842
axis labels, 858
central line, 845
control limit equations, 845–847

control limits, computing, 841
examples, advanced, 859
examples, introductory, 822
missing values, 859
notation, 845
ODS tables, 853
options summarized by function, 834–840
overview, 821
plotted points, 845
plotting character, 833
plotting subgroup means, 842
probability limits, 841
process mean, specifying, 842
process standard deviation, specifying, 844
reading preestablished control limit parameters, 830, 831, 854, 855
reading probability limits, 843
reading raw measurements, 822–824, 853, 854
reading subgroup summary statistics, 825–827, 855, 856
reading summary statistics and control limits, 830, 856
saving control limit parameters, 828, 829, 851
saving subgroup summary statistics, 827, 828, 852
saving summary statistics and control limits, 829, 830, 852, 853
span of moving average, choosing, 847
span parameter, specifying, 844
specifying parameters for, 859, 861
standard deviation, estimating, 857, 858
syntax, 832

V

V-mask charts,
See cumulative sum control charts
variance
divisors for, 183
saving in output data set, 399
variance components method
GAGE application, 2069
gage studies, 2059, 2078
variance of median,
See STDMED function
variation, multiple components of, 2009, 2010, 2012–2016
determining components, 2013–2016
preliminary examination, 2010, 2012, 2013
VBAR charts
options summarized by function, 971–976
syntax, 970

W

Weibull distribution
cdf plots, 241
chi-square goodness-of-fit test, 323
deviation from empirical distribution, 323
EDF goodness-of-fit test, 323
histograms, 312, 319, 347
P-P plots, 422
probability plots, 452, 453
Q-Q plots, 483–485
weighted Pareto charts, 1050
Western Electric rules,
See Shewhart charts, tests for special causes
Wilcoxon signed rank test, 194
Winsorized means, 183, 200

X

$\bar{X}$ and R charts
axis labels, 1776
capability indices, computing, 1765, 1774, 1775
capability indices, saving, 1746
central line, 1762
control limit equations, 1763
examples, advanced, 1777
examples, introductory, 1738
missing values, 1776
notation, 1762
ODS tables, 1767
options summarized by function, 1751–1756, 1758, 1759
overview, 1737
plotted points, 1762
plotting character, 1751
reading preestablished control limits, 1748, 1768
reading raw measurements, 1738, 1767
reading subgroup summary statistics, 1741, 1768, 1769
reading summary statistics and control limits, 1747, 1748, 1769, 1770
saving control limits, 1745, 1764, 1765
saving subgroup summary statistics, 1744, 1765
saving summary statistics and control limits, 1746, 1747, 1766
specifying parameters for, 1780, 1781
standard deviation, estimating, 1773
syntax, 1750
tests for special causes, 1777, 1778
$\bar{X}$ and s charts
ODS tables, 1819
$\bar{X}$ and s charts
capability indices, computing, 1817
central line, 1814
control limit equations, 1815
examples, advanced, 1826
examples, introductory, 1790
labeling axes, 1825
missing values, 1825
notation, 1814
options summarized by function, 1802–1805, 1807–1809, 1811
overview, 1789
plotted points, 1814
reading preestablished control limits, 1800, 1820
reading raw measurements, 1790–1792, 1819
reading subgroup summary statistics, 1793, 1794, 1796, 1820, 1821

reading summary statistics and control limits, 1798, 1821, 1822
saving control limits, 1797, 1816, 1817
saving subgroup summary statistics, 1796, 1817
saving summary statistics and control limits, 1797, 1798, 1818
specifying parameters for, 1815
standard deviation, estimating, 1823, 1824
syntax, 1801
$\bar{X}$ and s charts
plotting character, 1802
$\bar{X}$ charts
axis labels, 1725
capability indices, computing, 1717
central line, 1715
control limit equations, 1715, 1716
examples, advanced, 1726
examples, introductory, 1692
missing values, 1726
notation, 1714
OC curves, 1731, 1732
ODS tables, 1719
options summarized by function, 1704, 1705, 1707–1711, 1714
overview, 1691
plotted points, 1715
plotting character, 1704
reading preestablished control limits, 1701–1703, 1720, 1721
reading raw measurements, 1692–1694, 1719, 1720
reading subgroup summary statistics, 1694–1697, 1721, 1722
reading summary statistics and control limits, 1700, 1701, 1722, 1723
saving control limits, 1699, 1716, 1717, 1732
saving subgroup summary statistics, 1697–1699, 1717, 1718
saving summary statistics and control limits, 1700, 1701, 1718, 1719
standard deviation, estimating, 1723–1725, 1728–1730
syntax, 1703
tests for special causes, 1726, 1727

Syntax Index

A

ALLLABEL2= option
  CUSUM procedure, 1854
  MACONTROL procedure, 1854
  SHEWHART procedure, 1854
ALLLABEL= option
  CUSUM procedure, 1854
  MACONTROL procedure, 1854
  SHEWHART procedure, 1854
ALLN option
  CUSUM procedure, 1854
  MACONTROL procedure, 1854
  SHEWHART procedure, 1854, 1981
ALPHA= option
  CUSUM procedure, 544
  MACONTROL procedure, 786
  SHEWHART procedure, 1855
ANCHOR= option
  PARETO procedure, 977, 1010
ANGLE= option
  PARETO procedure, 977, 1011
ANNOKEY option
  PARETO procedure, 977, 1011
ANNOTATE2= data set
  PARETO procedure, 959
ANNOTATE2= option
  CUSUM procedure, 1855
  MACONTROL procedure, 1855
  SHEWHART procedure, 1855
ANNOTATE= data set
  PARETO procedure, 959
ANNOTATE= option
  CUSUM procedure, 1855
  MACONTROL procedure, 1855
  SHEWHART procedure, 1855
ANOM procedure, 8
  syntax, 8
ANOM procedure, BOXCHART statement,
  See also ANOM procedure, all chart statements
  ALPHA= option, 129
  BOX= data set, 133
  DATA= data set, 135
  LIMITN= option, 129
  LIMITS= data set, 135, 136
  MEAN= option, 129
  missing values, 139
  MSE= option, 129
  NOCHART option, 117, 118
  OUTBOX= data set, 130
  OUTLIMITS= data set, 118, 119, 131
  OUTSUMMARY= data set, 117, 118, 131, 132
  OUTTABLE= data set, 119, 120, 133
  SUMMARY= data set, 114–116, 136, 138
  TABLE= data set, 119, 120, 138
ANOM procedure, INSET statement
  CFILL= option, 150
  CFILLH= option, 151
  CFRAME= option, 151
  CHEADER= option, 151
  CSHADOW= option, 151
  CTEXT= option, 151
  DATA option, 151
  FONT= option, 151
  FORMAT= option, 151
  HEADER= option, 151
  HEIGHT= option, 152
  NOFRAME option, 152
  POSITION= option, 152–154
  REFPOINT= option, 152
ANOM procedure, PCHART statement,
  See also ANOM procedure, all chart statements
  ALPHA= option, 74
  DATA= data set, 76, 77
  GROUPN= option, 57
  LIMITN= option, 74
  LIMITS= data set, 77, 78
  missing values, 80
  OUTLIMITS= data set, 61, 62, 74, 75
  OUTSUMMARY= data set, 60, 75
  OUTTABLE= data set, 62, 63, 76
  P= option, 74
  SUMMARY= data set, 59, 78, 79
  TABLE= data set, 63, 79
ANOM procedure, UCHART statement,
  See also ANOM procedure, all chart statements
  ALPHA= option, 101
  DATA= data set, 103
  GROUPN= option, 87
  LIMITN= option, 101
  LIMITS= data set, 104
  missing values, 106
  NOCHART option, 88, 89
  OUTLIMITS= data set, 88, 101
  OUTSUMMARY= data set, 102
  OUTTABLE= data set, 89, 90, 102, 103
  SUMMARY= data set, 105

TABLE= data set, 90, 105, 106
U= option, 101
ANOM procedure, XCHART statement,
See also ANOM procedure, all chart statements
ALPHA= option, 32, 34
DATA= data set, 37, 38
LIMITN= option, 32, 34
LIMITS= data set, 38, 39
MEAN= option, 32, 34
missing values, 41
MSE= option, 32, 34
NOCHART option, 20, 21
OUTLIMITS= data set, 21, 22, 35
OUTSUMMARY= data set, 20, 21, 36
OUTTABLE= data set, 22, 23, 37
SUMMARY= data set, 18–20, 39, 40
TABLE= data set, 22, 23, 40
AOQ2 function, 2092, 2093, 2115
ASN2 function, 2093, 2095, 2115
ASYMPTOTIC option
MACONTROL procedure, 787, 842
ATI2 function, 2095, 2096, 2115
AXISFACTOR option
PARETO procedure, 978, 1011

B

BARLABEL= option
PARETO procedure, 978, 1012
BARLABPOS= option
PARETO procedure, 978, 1012
BARLEGEND= option
PARETO procedure, 978, 1012
BARLEGLABEL= option
PARETO procedure, 979, 1013
BARWIDTH= option
PARETO procedure, 979, 1013
BAYESACT call, 2096–2098
BETA= option
CUSUM procedure, 545
BILEVEL option
CUSUM procedure, 1856
MACONTROL procedure, 1856
SHEWHART procedure, 1856
block-variables, ANOM procedure
BOXCHART statement, 120
PCHART statement, 64
UCHART statement, 91
XCHART statement, 23
block-variables, CUSUM procedure
XCHART statement, 536
block-variables, MACONTROL procedure
EWMACHART statement, 777
MACHART statement, 833
block-variables, SHEWHART procedure
BOXCHART statement, 1253
CCHART statement, 1316
displaying values, 1858
IRCHART statement, 1357
labels, 1856, 1857
legends, 1857, 1861
MCHART statement, 1405
MRCHART statement, 1449
NPCHART statement, 1494
PCHART statement, 1538
RCHART statement, 1585
SCHART statement, 1625
UCHART statement, 1663
XCHART statement, 1704
XRCHART statement, 1750
XSCHART statement, 1801
BLOCKLABELPOS= option
CUSUM procedure, 1856
MACONTROL procedure, 1856
SHEWHART procedure, 1856, 1936, 2024
BLOCKLABTYPE= option
CUSUM procedure, 1857
MACONTROL procedure, 1857
SHEWHART procedure, 1857, 2024
BLOCKPOS= option
CUSUM procedure, 1857
MACONTROL procedure, 1857
SHEWHART procedure, 1857, 1934, 1936
BLOCKREP option
CUSUM procedure, 1858
MACONTROL procedure, 1858
SHEWHART procedure, 1858
BLOCKS statement, FACTEX procedure,
See FACTEX procedure, BLOCKS statement
options summarized by function, 602
syntax, 606
BLOCKS statement, OPTEX procedure,
See OPTEX procedure, BLOCKS statement
syntax, 893
BOXCHART statement,
See also SHEWHART procedure, BOXCHART statement
examples, advanced, 1284
examples, introductory, 1240
options summarized by function, 1254, 1256, 1258, 1259, 1261, 1263
overview, 1239
syntax, 1252
BOXCHART statement, ANOM procedure,
See also ANOM procedure, BOXCHART statement
examples, advanced, 139
examples, introductory, 111
options summarized by function, 121–125, 127
overview, 111
syntax, 120
BOXCONNECT option
SHEWHART procedure, 1858
BOXSTYLE= option
SHEWHART procedure, 1298, 1858
BOXSTYLE= option, SHEWHART procedure, 1302
BOXWIDTH= option
SHEWHART procedure, 1861
BOXWIDTHSCALE= option

SHEWHART procedure, 1861

C

C4 function, 2098, 2099, 2115
CAPABILITY procedure, 170
and PROC SHEWHART, 2028, 2031
introduction, 161
syntax, 170
CAPABILITY procedure, CDFPLOT statement
ALPHA= beta-option, 233
ALPHA= gamma-option, 233
ALPHADELTA= gamma-option, 233
ALPHAINITIAL= gamma-option, 233
ANNOTATE= option, 233
BETA beta-option, 233
BETA= option, 234
C= option, 234
CAXIS= option, 234
CDELTA= option, 234
CDFSYMBOL= option, 234
CFRAME= option, 235
CHREF= option, 235
CINITIAL= option, 235
COLOR= option, 235
CTEXT= option, 235
CVREF= option, 235
DESCRIPTION= option, 235
EXPONENTIAL option, 235
FONT= option, 236
GAMMA option, 236
HAXIS= option, 236
HMINOR= option, 237
HREF= option, 237
HREFCHAR= option, 237
HREFLABELS= option, 237
LEGEND= option, 237
LHREF= option, 237
LOGNORMAL option, 237
LVREF= option, 238
MAXITER= option, 238
MU= option, 238
NAME= option, 238
NOCDFLEGEND option, 238
NOECDF option, 238
NOFRAME option, 238
NOLEGEND option, 238
NORMAL option, 239
NOSPECLEGEND option, 239
SCALE= option, 239
SHAPE= option, 239
SIGMA= option, 239
SYMBOL= option, 240
THETA= option, 240
THRESHOLD= option, 240
VAXIS= option, 240
VMINOR= option, 240
VREF= option, 240
VREFCHAR= option, 240
VREFLABELS= option, 240
VSCALE= option, 240
W= option, 240
WEIBULL Weibull-option, 241
ZETA= option, 241
CAPABILITY procedure, COMPHISTOGRAM statement
ANNOKEY option, 257
ANNOTATE= option, 257
BARLABEL= option, 257
BARWIDTH= option, 257
C= option, 257
CAXIS= option, 258
CBARLINE= option, 258
CFILL= option, 258
CFRAME= option, 258
CFRAMENLEG= option, 258
CFRAMESIDE= option, 258
CFRAMETOP= option, 259
CGRID= option, 259
CHREF= option, 259
CLASS= option, 252
CLASSKEY= option, 259
CLASSSPECS= option, 260
CLIPSPEC= option, 261
COLOR= option, 261
CPROP= option, 261
CTEXT= option, 261
CTEXTSIDE= option, 261
CTEXTTOP= option, 262
CVREF= option, 262
DESCRIPTION= option, 262
ENDPOINTS= option, 262, 297
FILL option, 262, 263
FONT= option, 263
FRONTREF option, 263
GRID option, 263
HEIGHT= option, 263
HOFFSET= option, 263
HREF= option, 263
HREFLABELS= option, 263
HREFLABPOS= option, 264
INFONT= option, 264
INHEIGHT= option, 264
INTERTILE= option, 264
K= option, 264
KERNEL kernel-option, 257, 264
L= option, 265
LGRID= option, 265
LHREF= option, 265
LOWER= option, 265
LVREF= option, 265
MAXNBIN= option, 265
MAXSIGMAS= option, 265
MIDPOINTS= option, 266
MISSING1 option, 266
MISSING2 option, 266
MU= option, 266
NAME= option, 266
NCOLS= option, 267

NLEGEND option, 258, 267
NLEGENDPOS option, 267
NOBARS option, 267
NOCHART option, 267
NOFRAME option, 267
NOHLABEL option, 267
NOKEYMOVE option, 267
NOPLOT option, 268
NORMAL normal-option, 268
NOVLABEL option, 268
NOVTICK option, 268
NROWS= option, 268
ORDER1= option, 268
ORDER2= option, 269
OUTHISTOGRAM= option, 270
PFILL= option, 270
RTINCLUDE option, 270
SIGMA= option, 270
TILELEGLABEL= option, 270
TURNVLABELS option, 270
UPPER= option, 270
VAXIS= option, 270
VAXISLABEL= option, 271
VOFFSET= option, 271
VREF= option, 271
VREFLABELS= option, 271
VREFLABPOS= option, 271
VSCALE= option, 271
W= option, 271
WAXIS= option, 271
WBARLINE= option, 271
WGRID= option, 271

CAPABILITY procedure, HISTOGRAM statement
ALPHA= option, 292, 313
ALPHADELTA= gamma-option, 292
ALPHAINITIAL= gamma-option, 292
ANNOTATE= option, 292
BARLABEL= option, 292
BETA beta-option, 292, 313
BETA= option, 293, 313
BMCBOXFILL= option, 293
BMCFRAME= option, 293
BMCOLOR= option, 293
BMMARGIN= option, 294
BMPLOT= option, 294
C= option, 294, 319, 320
CAXIS= option, 295
CBARLINE= option, 295
CDELTA= option, 295
CFILL= option, 295
CFRAME= option, 295
CHREF= option, 295
CINITIAL= Weibull-option, 296
CLIPSPEC= option, 296
COLOR= option, 296
CTEXT= option, 296
CURVELEGEND= option, 296
CVREF= option, 296
DELTA= option, 296, 315, 317
DESCRIPTION= option, 296
EXPONENTIAL exponential-option, 297, 314
FILL option, 297, 298
FITINTERVAL= option, 298
FITMETHOD= option, 298
FITTOLERANCE= option, 298
FONT= option, 298
FORCEHIST option, 298
FRONTREF option, 298
GAMMA gamma-option, 299, 315
GAMMA= option, 299, 315, 317
HANGING option, 299
HAXIS= option, 300
HMINOR= option, 300
HREF= option, 300
HREFCHAR= option, 301
HREFLABELS= option, 301
INDICES option, 301, 325, 327
K= option, 301, 319, 320
KERNEL option, 301, 319, 320
L= option, 302
LEGEND= option, 302
LHREF= option, 302
LOGNORMAL lognormal-option, 302, 318
LVREF= option, 303
MAXITER= option, 303
MIDPERCENTS option, 303, 328
MIDPOINTS= option, 304
MIDPTAXIS= option, 305
MU= option, 305, 319
NAME= option, 305
NENDPOINTS= option, 305
NMIDPOINTS= option, 305
NOBARS option, 305
NOCURVELEGEND option, 305
NOFRAME option, 306
NOLEGEND option, 306
NOPLOT option, 306
NOPRINT option, 306
NORMAL normal-option, 306, 319
NOSPECLEGEND option, 306
OUTFIT= option, 307, 328
OUTHISTOGRAM= option, 307, 328, 330, 331
PCTAXIS= option, 307
PERCENTS= option, 307, 328
PFILL= option, 307
RTINCLUDE option, 308
SB S_B-option, 315
SB S_U-option, 317
SB S_B-option, 308
SCALE= option, 309, 314, 315, 319
SHAPE= option, 309, 318, 319
SIGMA= option, 309, 313, 315, 317, 319
SPECLEGEND= option, 309
SU S_U-option, 309
SYMBOL= option, 310
THETA= option, 310, 313
THRESHOLD= option, 310, 315, 317–319
VAXIS= option, 311

VMINOR= option, 311
VREF= option, 311
VREFCHAR= option, 311
VREFLABELS= option, 311
VSCALE= option, 311
W= option, 311
WBARLINE= option, 311
WEIBULL option, 312, 319
ZETA= option, 312
CAPABILITY procedure, INSET statement
CFILL= option, 368
CFILLH= option, 369
CFRAME= option, 369
CHEADER= option, 369
CSHADOW= option, 369
CTEXT= option, 369
DATA option, 369
displaying C_{pk}, 500
FONT= option, 369
FORMAT= option, 369
HEADER= option, 369
HEIGHT= option, 370
NOFRAME option, 370
POSITION= option, 370–372
REFPOINT= option, 370
CAPABILITY procedure, INTERVALS statement
ALPHA= option, 384
K= option, 384
METHODS= option, 385–388
NOPRINT option, 385
OUTINTERVALS= option, 385, 389
P= option, 385
TYPE= option, 386
CAPABILITY procedure, OUTPUT statement
OUT= option, 396, 401
PCTLNAME= option, 398
PCTLPRE= option, 397
PCTLPTS= option, 396
CAPABILITY procedure, PPPLOT statement
ALPHA= option, 415, 418
ANNOTATE= option, 415
BETA option, 413, 415
BETA= option, 416
C= option, 416, 422
CAXIS= option, 416
CFRAME= option, 416
CHREF= option, 416
COLOR= option, 411, 416
CTEXT= option, 416
CVREF= option, 416
DESCRIPTION= option, 417
EXPONENTIAL option, 413, 417
FONT= option, 417
GAMMA option, 413, 417
HAXIS= option, 418
HMINOR= option, 418
HREF= option, 418
HREFCHAR= option, 418
HREFLABELS= option, 419
L= option, 419
LHREF= option, 419
LOGNORMAL option, 413, 419
LVREF= option, 420
MU= option, 413, 420, 421
NAME= option, 420
NOFRAME option, 420
NOLINE option, 420
NOOBSLEGEND option, 420
NORMAL option, 413, 420
PPSYMBOL= option, 421
SCALE= option, 418, 420, 421
SHAPE= option, 418, 420, 421
SIGMA= option, 413, 418, 419, 421, 422
SQUARE option, 411, 421
SYMBOL= option, 421
THETA= option, 418, 419, 421, 422
THRESHOLD= option, 418, 420, 421
VAXIS= option, 421, 423
VMINOR= option, 422
VREF= option, 422
VREFCHAR= option, 422
VREFLABELS= option, 422
W= option, 422
WEIBULL option, 413, 422
ZETA= option, 419, 423
CAPABILITY procedure, PROBPLOT statement
ALPHA= option, 441
ANNOTATE= option, 441
BETA option, 439, 441
BETA= option, 443
C= option, 443, 452, 454
CAXIS= option, 443
CFRAME= option, 443
CHREF= option, 443
COLOR= option, 443
CTEXT= option, 444
CVREF= option, 444
DESCRIPTION= option, 444
EXPONENTIAL option, 439, 444
FONT= option, 444
GAMMA option, 439, 445
GRID option, 446
GRIDCHAR= option, 446
HAXIS= option, 446
HMINOR= option, 446
HREF= option, 446, 460
HREFCHAR= option, 446
HREFLABELS= option, 446, 460
L= option, 446
LEGEND= option, 446
LGRID= option, 446
LHREF= option, 446, 460
LOGNORMAL option, 439, 447
LVREF= option, 448, 460
MU= option, 448, 449
NADJ= option, 448, 453
NAME= option, 448
NOFRAME option, 448

NOLEGEND option, 448
NOLINELEGEND option, 448
NOOBSLEGEND option, 448
NORMAL option, 439, 448
NOSPECLEGEND option, 449
PCTLMINOR option, 449, 460
PCTLORDER= option, 449
PROBSYMBOL option, 449
RANKADJ= option, 450, 453
ROTATE option, 450
SCALE= option, 444, 446, 450, 453
SHAPE= option, 450, 452
SIGMA= option, 442, 449, 450, 454
SLOPE= option, 451
SQUARE option, 451, 460
SYMBOL= option, 451
THETA= option, 442, 447, 451
THRESHOLD= option, 444, 446, 451, 453
VAXIS= option, 452, 458
VMINOR= option, 452
VREF= option, 452
VREFCHAR= option, 452
VREFLABELS= option, 452
W= option, 452
WEIBULL option, 439, 452
WEIBULL2 option, 439, 453
ZETA= option, 447, 454
CAPABILITY procedure, PROC CAPABILITY statement
ALL option, 176
ALPHA= option, 176–178, 183, 222, 1864
ANNOTATE= option, 176, 189
CHECKINDICES option, 176
CIBASIC= option, 177
CIINDICES= option, 177
CIPCTLDF= option, 177
CIPCTLNORMAL= option, 178
CIPROBEX option, 178
CIQUANTDF= option, 177
CIQUANTNORMAL= option, 178
CPMA= option, 178, 182
DATA= option, 178, 187
DEF= option, 178, 181
EXCLNPWGT option, 178
FORMCHAR= option, 179
FREQ option, 180
GOUT= option, 180
LINEPRINTER option, 180
LOCATION= option, 180
LOCCOUNT option, 180
missing values, 215
MODE option, 180
MODES option, 180, 203
MUO= option, 180
NEXTROBS= option, 180
NEXTRVAL= option, 180
NOPRINT option, 180
NORMALTEST option, 181, 194
ODS tables, 216
OUTTABLE= option, 181, 190
PCTLDEF= option, 178, 181, 197
ROBUSTSCALE option, 181, 200
ROUND= option, 181
SPEC= option, 182, 188
SPECIALINDICES option, 182
TRIM option, 182
TRIMMED option, 182
TRIMMED= option, 200
TYPE= option, 177, 178, 182, 183, 1864
VARDEF= option, 183
WINSOR option, 183
WINSORIZED option, 183
WINSORIZED= option, 200
CAPABILITY procedure, QQPLOT statement
ALPHA= option, 471, 474
ANNOTATE= option, 471
BETA option, 468, 469, 471
BETA= option, 472
C= option, 473, 483, 485
CAXIS= option, 473
CFRAME= option, 473
CHREF= option, 473
COLOR= option, 466, 468, 473
CPKREF option, 473, 478, 500
CPKSCALE option, 473, 478, 500
CTEXT= option, 473
CVREF= option, 474
DESCRIPTION= option, 474
EXPONENTIAL option, 468, 469, 474
FONT= option, 474
GAMMA option, 468, 469, 474
GRID option, 479
GRIDCHAR= option, 479
HAXIS= option, 475
HMINOR= option, 475
HREF= option, 475
HREFCHAR= option, 476
HREFLABELS= option, 476
L= option, 466, 476
LABEL= option, 479
LEGEND= option, 476
LGRID= option, 479
LHREF= option, 476
LOGNORMAL option, 468, 469, 476
LVREF= option, 477
MU= option, 466, 468, 477, 478
NADJ= option, 477, 486
NAME= option, 477
NOFRAME option, 477
NOLEGEND option, 477
NOLINELEGEND option, 478
NOOBSLEGEND option, 478
NORMAL option, 468, 469, 478, 500
NOSPECLEGEND option, 466, 479
PCTLAXIS option, 479, 489
PCTLMINOR option, 480
PCTLSCALE option, 480, 489
QQSYMBOL= option, 480

RANKADJ= option, 481, 486
ROTATE option, 481
SCALE= option, 472, 474, 475, 477, 481, 484
SHAPE= option, 474, 476, 481, 483
SIGMA= option, 466, 468, 472, 474–476, 478, 481, 484, 485
SLOPE= option, 477, 481, 485
SQUARE option, 466, 482
SYMBOL= option, 482
THETA= option, 472, 474, 475, 477, 482, 484
THRESHOLD= option, 472, 474, 475, 477, 482, 484
VAXIS= option, 482
VMINOR= option, 482
VREF= option, 483
VREFCHAR= option, 483
VREFLABELS= option, 483
W= option, 483
WEIBULL option, 468, 469, 483
WEIBULL2 option, 468, 469, 484
ZETA= option, 477, 485
CAPABILITY procedure, SPEC statement
CLEFT= option, 185
CLSL= option, 185
CRIGHT= option, 185
CTARGET= option, 185
CUSL= option, 185
LLSL= option, 185
LSL= option, 185
LSLSYMBOL= option, 186
LTARGET= option, 186
LUSL= option, 186
PLEFT= option, 186
PRIGHT= option, 186
TARGET= option, 186
TARGETSYMBOL= option, 186
USL= option, 186
USLSYMBOL= option, 187
WLSL= option, 187
WTARGET= option, 187
WUSL= option, 187
CATLEGLABEL= option
PARETO procedure, 979, 1013
CAXIS2= option
PARETO procedure, 980, 1013
CAXIS= option
CUSUM procedure, 1861
MACONTROL procedure, 1861
PARETO procedure, 979, 1013
SHEWHART procedure, 1861
CBARLINE= option
PARETO procedure, 980, 1013
CBARS= option
PARETO procedure, 980, 1014
CBLOCKLAB= option
CUSUM procedure, 1861
MACONTROL procedure, 1861
SHEWHART procedure, 1861
CBLOCKVAR= option
CUSUM procedure, 1861
MACONTROL procedure, 1861
SHEWHART procedure, 1861, 1934, 1936
CBOXES= option
SHEWHART procedure, 1862
CBOXFILL= option
SHEWHART procedure, 1862
CCHART statement, SHEWHART procedure,
See also SHEWHART procedure, CCHART statement
examples, advanced, 1337
examples, introductory, 1306
options summarized by function, 1317, 1318, 1320, 1321, 1324
overview, 1305
syntax, 1315
CCLIP= option
MACONTROL procedure, 1863
SHEWHART procedure, 1863
CCONNECT= option
CUSUM procedure, 1863
MACONTROL procedure, 1863
PARETO procedure, 980, 1014
SHEWHART procedure, 1863
CCOVERLAY2= option
SHEWHART procedure, 1863
CCOVERLAY= option
SHEWHART procedure, 1863
CDFPLOT statement,
See CAPABILITY procedure, CDFPLOT statement
examples, 242, 243
getting started, 227
options summarized by function, 229–232
overview, 227
syntax, 228
CFRAME= option
CUSUM procedure, 1863
MACONTROL procedure, 1863
PARETO procedure, 980, 1014
SHEWHART procedure, 1863, 1937
CFRAMELAB= option
CUSUM procedure, 1863
MACONTROL procedure, 1863
SHEWHART procedure, 1863
CFRAMENLEG= option
PARETO procedure, 981, 1014
CFRAMESIDE= option
PARETO procedure, 981, 1014
CFRAMETOP= option
PARETO procedure, 981, 1015
CGRID2= option
PARETO procedure, 981, 1015
CGRID= option
CUSUM procedure, 1863
MACONTROL procedure, 1863
PARETO procedure, 981, 1015
SHEWHART procedure, 1863
character subgroup variables

SHEWHART procedure, 1882
CHIGH(*n*)= option
PARETO procedure, 981, 1015
CHREF= option
CUSUM procedure, 1864
MACONTROL procedure, 1864
PARETO procedure, 981, 1015
SHEWHART procedure, 1864
CINFILL= option
CUSUM procedure, 545
MACONTROL procedure, 1864
SHEWHART procedure, 1864
CLABEL= option
CUSUM procedure, 1864
MACONTROL procedure, 1864
SHEWHART procedure, 1864
CLASS statement, OPTEX procedure,
See OPTEX procedure, CLASS statement
syntax, 895
CLASS= option
PARETO procedure, 981, 1015
CLASSKEY= option
PARETO procedure, 982, 1016
CLIMITS= option
CUSUM procedure, 545
MACONTROL procedure, 1864
SHEWHART procedure, 1864
CLIPCHAR= option
MACONTROL procedure, 1864
SHEWHART procedure, 1864
CLIPFACTOR= option
MACONTROL procedure, 1864
SHEWHART procedure, 1864, 1963, 1965
CLIPLEGEND= option
MACONTROL procedure, 1865
SHEWHART procedure, 1865, 1965
CLIPLEGPOS= option
MACONTROL procedure, 1865
SHEWHART procedure, 1865, 1965
CLIPSUBCHAR= option
MACONTROL procedure, 1865
SHEWHART procedure, 1865, 1965
CLIPSYMBOL= option
MACONTROL procedure, 1865
SHEWHART procedure, 1865, 1965
CLIPSYMBOLHT= option
SHEWHART procedure, 1865
CLOW(*n*)= option
PARETO procedure, 983, 1016
CMASK= option
CUSUM procedure, 545
CMEANSYMBOL= option
MACONTROL procedure, 787, 842
CMPCTLABEL option
PARETO procedure, 983, 1016
CNEEDLES= option
CUSUM procedure, 1865
MACONTROL procedure, 1865
SHEWHART procedure, 1865, 1961
COMPHISTOGRAM statement,
See CAPABILITY procedure,
COMPHISTOGRAM statement
examples, 248, 249
getting started, 247
options summarized by function, 253, 254, 256,
257
overview, 247
syntax, 251
CONNECTCHAR= option
CUSUM procedure, 1866
MACONTROL procedure, 1866
PARETO procedure, 983
SHEWHART procedure, 1866
CONTROLSTAT= option
SHEWHART procedure, 1866
COTHER= option
PARETO procedure, 983, 1016
COUT= option
CUSUM procedure, 1866
MACONTROL procedure, 1866
SHEWHART procedure, 1866
COUTFILL= option
CUSUM procedure, 1866
MACONTROL procedure, 1866
SHEWHART procedure, 1866
COVERLAY2= option
SHEWHART procedure, 1866
COVERLAY= option
SHEWHART procedure, 1866
COVERLAYCLIP= option
SHEWHART procedure, 1866
CPHASEBOX= option
SHEWHART procedure, 1298, 1866
CPHASEBOXCONNECT= option
SHEWHART procedure, 1866
CPHASEBOXFILL= option
SHEWHART procedure, 1298, 1867
CPHASELEG= option
CUSUM procedure, 1867
MACONTROL procedure, 1867
SHEWHART procedure, 1867, 1937
CPHASEMEANCONNECT= option
SHEWHART procedure, 1298, 1867
CPMA= option
CAPABILITY procedure, 182
CPROP= option
PARETO procedure, 983, 1017
CSTARCIRCLES= option
CUSUM procedure, 1867
MACONTROL procedure, 1867
SHEWHART procedure, 1867
CSTARFILL= option
CUSUM procedure, 1867
MACONTROL procedure, 1867
SHEWHART procedure, 1867
CSTAROUT= option
CUSUM procedure, 1867
MACONTROL procedure, 1867

SHEWHART procedure, 1867
CSTARS= option
CUSUM procedure, 1868
MACONTROL procedure, 1868
SHEWHART procedure, 1868
CSYMBOL= option
SHEWHART procedure, 1868
CTESTLABBOX= option
SHEWHART procedure, 1869
CTESTS= option
SHEWHART procedure, 1869, 1990
CTESTSYMBOL= option
SHEWHART procedure, 1869
CTEXT= option
CUSUM procedure, 1869
MACONTROL procedure, 1869
PARETO procedure, 983, 1017
SHEWHART procedure, 1869
CTEXTSIDE= option
PARETO procedure, 983, 1017
CTEXTTOP= option
PARETO procedure, 983, 1017
CTILES= option
PARETO procedure, 984, 1017
CUSUM procedure, 514
ANNOTATE2= option, 514
ANNOTATE= option, 514
DATA= data set, 514
FORMCHAR= option, 514, 515
GOUT= option, 515
GRAPHICS option, 532
HISTORY= data set, 515, 516
introduction, 509
LIMITS= data set, 516
LINEPRINTER option, 516
overview, 513
syntax, 514
CUSUM procedure, XCHART statement
ALLN option, 553
ALPHA= option, 522, 535, 544, 556
BETA= option, 545, 556
CINFILL= option, 545
CLIMITS= option, 545
CMASK= option, 545
DATA= data set, 522, 523, 566, 567
DATAUNITS option, 545, 553
DELTA= option, 522, 535, 545, 551
H= option, 529, 530, 535, 546, 556
HEADSTART= option, 546, 553
HISTORY= data set, 525, 526, 568, 569
INTERVAL= option, 555
K= option, 529, 530, 546, 556
LIMITN= option, 547, 551, 553
LIMITS= data set, 533, 534, 567, 568
LLIMITS= option, 547
LMASK= option, 547
missing values, 569
MU0= option, 522, 535, 547, 551
NOARL option, 547
NOMASK option, 547
NOREADLIMITS option, 547
ORIGIN= option, 547
OUTHISTORY= data set, 527, 528, 565
OUTLIMITS= data set, 531, 532, 564
OUTTABLE= data set, 565, 569, 572
READINDEX= option, 548
READLIMITS option, 548
READSIGMAS option, 548
SCHEME= option, 529, 530, 535, 549
SHIFT= option, 549, 551
SIGMA0= option, 522, 549
SIGMAS= option, 549
SMETHOD= option, 549, 561–563
TABLEALL option, 529, 530, 550
TABLECHART option, 550
TABLECOMP option, 550
TABLEID option, 550
TABLEOUT option, 550
TABLESUMMARY option, 550
TYPE= option, 550, 551
VAXIS= option, 522
WLIMITS= option, 550
WMASK= option, 550
CUSUMARL function, 2099, 2101
CVREF= option
CUSUM procedure, 1869
MACONTROL procedure, 1869
PARETO procedure, 984, 1017
SHEWHART procedure, 1869
CZONES= option
SHEWHART procedure, 1870, 1990

D

D2 function, 2101, 2115
D3 function, 2102, 2115
DATA= data set
PARETO procedure, 959
DATAUNIT= option
SHEWHART procedure, 1870
DATAUNITS option
CUSUM procedure, 545
DELTA= option
CUSUM procedure, 545
DESCENDING option
CLASS statement (OPTEX), 895
DESCRIPTION2= option
SHEWHART procedure, 1870
DESCRIPTION= option
CUSUM procedure, 1870
MACONTROL procedure, 1870
PARETO procedure, 984, 1018
SHEWHART procedure, 1870

E

ENDGRID option
CUSUM procedure, 1870
MACONTROL procedure, 1870
SHEWHART procedure, 1870

EWMAARL function, 2103
EWMACHART statement,
See also MACONTROL procedure, EWMACHART statement
examples, advanced, 805
examples, introductory, 766
overview, 765
syntax, 777
EXAMINE statement, FACTEX procedure,
See FACTEX procedure, EXAMINE statement
options summarized by function, 603
syntax, 608
EXAMINE statement, OPTEX procedure,
See OPTEX procedure, EXAMINE statement
syntax, 900
EXCHART option
CUSUM procedure, 1870
MACONTROL procedure, 1870
SHEWHART procedure, 1870

F

FACTEX procedure, 601
getting started, 590
learning about FACTEX, 590
overview, 589
summary of functions, 601
syntax, 601
FACTEX procedure, BLOCKS statement
NBLKFACS= option, 606
NBLKFACS=MAXIMUM option, 607
NBLOCKS= option, 607
NBLOCKS= option, examples, 593, 635
NBLOCKS=MAXIMUM option, 607
SIZE= option, 607
SIZE=MINIMUM option, 607
FACTEX procedure, EXAMINE statement
ALIASING option, 608
ALIASING option, example, 596
CONFOUNDING option, 609
DESIGN option, 609
DESIGN option, example, 591
FACTEX procedure, FACTORS statement
example, 591
NLEV= option, 609
FACTEX procedure, MODEL statement
ESTIMATE= option, 610
ESTIMATE= option, examples, 619, 636
MINABS option, 611, 657
MINABS option, example, 634
MINABS option, limitation, 634
NONNEGLIGIBLE= option, 610
RESOLUTION= option, 610
RESOLUTION= option, examples, 595, 618, 622
RESOLUTION=MAX option, 610
RESOLUTION=MAX option, examples, 593, 626
FACTEX procedure, OUTPUT statement
CVALS= option, 612, 613, 650
CVALS= option, example, 623
decode design factors, 612
derived factors, 614
derived factors, examples, 629, 631
DESIGNREP= option, 613
DESIGNREP= option, examples, 625–629
NOVALRAN option, 614
NVALS= option, 612, 613, 650
NVALS= option, example, 623
OUT= option, 612
OUT= option, example, 623
POINTREP= option, 614
POINTREP= option, examples, 625–629
RANDOMIZE= option, 614
RANDOMIZE= option, examples, 617, 623
RANDOMIZE= option, NOVALRAN option, 614
RANDOMIZE= option, seed, 614
recode block factor, 613
recode block factor levels, examples, 594, 623
recode design factor levels, examples, 592, 595, 623
FACTEX procedure, PROC FACTEX statement
example, 591
NAMELEN option, 605
NOCHECK option, 606, 634, 667
ODS tables, 659
SECONDS= option, 606
TIME= option, 606, 634
FACTEX procedure, SIZE statement
DESIGN= option, 616
DESIGN= option, examples, 595, 618
DESIGN=MINIMUM option, 616
FRACTION= option, 616
FRACTION=MAXIMUM option, 616
NRUNFACS= option, 616
NRUNFACS=MINIMUM option, 616
FACTORS statement, FACTEX procedure,
See FACTEX procedure, FACTORS statement
options summarized by function, 601
syntax, 609
FONT= option
CUSUM procedure, 1870
MACONTROL procedure, 1870
PARETO procedure, 984, 1018
SHEWHART procedure, 1562, 1780, 1870
FORMCHAR= option
PARETO procedure, 959
FREQ= option
PARETO procedure, 984, 1018

G

GAGE application
average and range method, 2067
average chart, 2066
data set format, 2080
entering data, 2062, 2064, 2071
gage catalog, 2061
introduction to, 2059

invoking, 2061
missing data, 2068
range chart, 2064
reading data set, 2071
saving data, 2070
variance components method, 2069
GENERATE statement, OPTEX procedure,
See OPTEX procedure, GENERATE statement
default options, 901
syntax, 901
GOUT= option
PARETO procedure, 959
GRID option
CUSUM procedure, 1871
MACONTROL procedure, 1871
PARETO procedure, 984, 1018
SHEWHART procedure, 1871
GRID2 option
PARETO procedure, 985, 1018
group-variable, ANOM procedure
BOXCHART statement, 120
PCHART statement, 64
UCHART statement, 91
XCHART statement, 23

H

H= option
CUSUM procedure, 546
HAXIS2= option
PARETO procedure, 1018
HAXIS2LABEL= option
PARETO procedure, 1019
HAXIS= option
CUSUM procedure, 1871
MACONTROL procedure, 1871
PARETO procedure, 1018
SHEWHART procedure, 1871
HAXISLABEL= option
PARETO procedure, 1018
HEADSTART= option
CUSUM procedure, 546
HEIGHT= option
CUSUM procedure, 1871
MACONTROL procedure, 1871
PARETO procedure, 985, 1019
SHEWHART procedure, 1871
HISTOGRAM statement,
See CAPABILITY procedure, HISTOGRAM statement
getting started, 279
options summarized by function, 286–289
overview, 279
syntax, 285
HLLEGLABEL= option
PARETO procedure, 985, 1019
HMINOR= option
CUSUM procedure, 1872
MACONTROL procedure, 1872
SHEWHART procedure, 1872
HOFFSET= option
CUSUM procedure, 1872
MACONTROL procedure, 1872
PARETO procedure, 985, 1019
SHEWHART procedure, 1872
HREF2= option
CUSUM procedure, 1872
MACONTROL procedure, 1872
PARETO procedure, 1019
SHEWHART procedure, 1872
HREF2DATA= option
CUSUM procedure, 1873
MACONTROL procedure, 1873
SHEWHART procedure, 1873
HREF2LABELS= option
CUSUM procedure, 1873
MACONTROL procedure, 1873
PARETO procedure, 1019
SHEWHART procedure, 1873
HREF= option
CUSUM procedure, 1872
MACONTROL procedure, 1872
PARETO procedure, 985, 1019
SHEWHART procedure, 1872
HREFCHAR= option
CUSUM procedure, 1873
MACONTROL procedure, 1873
PARETO procedure, 985
SHEWHART procedure, 1873
HREFDATA= option
CUSUM procedure, 1873
MACONTROL procedure, 1873
SHEWHART procedure, 1873
HREFLABELS= option
CUSUM procedure, 1873
MACONTROL procedure, 1873
PARETO procedure, 985, 1019
SHEWHART procedure, 1873
HREFLABPOS= option
CUSUM procedure, 1873
MACONTROL procedure, 1873
PARETO procedure, 985, 1019
SHEWHART procedure, 1873
HTML2= option
SHEWHART procedure, 1874
HTML= option
CUSUM procedure, 1874
MACONTROL procedure, 1874
PARETO procedure, 985, 1019
SHEWHART procedure, 1874
HTML_LEGEND= option
CUSUM procedure, 1874
MACONTROL procedure, 1874
SHEWHART procedure, 1874

I

ID statement, OPTEX procedure,
See OPTEX procedure, ID statement
syntax, 904

IDCOLOR= option
SHEWHART procedure, 1875
IDCTEXT= option
SHEWHART procedure, 1875
IDFONT= option
SHEWHART procedure, 1875
IDHEIGHT= option
SHEWHART procedure, 1875
IDSYMBOL= option
SHEWHART procedure, 1875
INFONT= option
PARETO procedure, 986, 1020
INHEIGHT= option
PARETO procedure, 986, 1020
INSET and INSET2 statements,
See CUSUM procedure, INSET statement
See MACONTROL procedure, INSET statement
See SHEWHART procedure, INSET and INSET2 statements
list of options, 1844
overview, 1835
syntax, 1841
INSET statement,
See ANOM procedure, INSET statement
See CAPABILITY procedure, INSET statement
See PARETO procedure, INSET statement
getting started, 143, 355, 579, 865, 1033, 1836
keywords summarized by function, 149, 362, 363, 365, 367, 1040, 1842
list of options, 149, 367, 1040
overview, 143, 355, 579, 865, 1033
syntax, 148, 359, 581, 867, 1038
INTERBAR= option
PARETO procedure, 986, 1020
INTERTILE= option
PARETO procedure, 986, 1020
INTERVAL= option
CUSUM procedure, 1875
MACONTROL procedure, 1875
SHEWHART procedure, 1875
INTERVALS statement,
See CAPABILITY procedure, INTERVALS statement
getting started, 379
list of options, 384
overview, 379
syntax, 383
INTSTART= option
CUSUM procedure, 1876
MACONTROL procedure, 1876
SHEWHART procedure, 1876
IRCHART statement,
See also SHEWHART procedure, IRCHART statement
examples, advanced, 1382
examples, introductory, 1348
options summarized by function, 1358, 1359, 1361, 1363, 1365, 1366, 1368, 1369
overview, 1347
syntax, 1357
Ishikawa diagrams
adding arrows, 691–694
aligning arrows, 709–715
balancing arrows, 709–715
data collection, 715, 716
data presentation, 715, 716
deleting arrows, 702–704
detail, decreasing, 716–718
detail, increasing, 716–718
editing existing diagrams, 739, 740
editing labels, 694–697
exporting diagrams, 726, 727
fonts, modifying, 727, 728
highlighting arrows, 729, 731–735
isolating arrows, 720, 721
labeling arrows, 694–697
managing complexity, 716–723
merging diagrams, 721–723
moving arrows, 697–702, 707–715
notepads, 715, 716
output, bitmaps, 726, 727
output, graphics, 724, 725
output, SAS data set, 738, 742–744
overview, 675
printing, bitmaps, 726, 727
printing, SAS/GRAPH output, 724, 725
resizing arrows, 704–707
SAS data set, input, 739, 740, 742–744
SAS data set, output, 738, 742–744
saving, bitmaps, 726, 727
saving, clipboard graphics, 726, 727
saving, graphics, 724, 725
saving, SAS data set, 738
subsetting arrows, 704–707, 729, 731–735
summary of operations, 689–691
swapping arrows, 707–709
tagging arrows, 704–707, 729, 731–735
terminology, 677
text entry, 694–697
undo, 702–704
zooming arrows, 719, 737
ISHIKAWA procedure, 744
syntax, 744

K

K= option
CUSUM procedure, 546

L

LABELANGLE= option
MACONTROL procedure, 1876
SHEWHART procedure, 1876
LABELFONT= option
MACONTROL procedure, 1876
SHEWHART procedure, 1876, 1956, 1957
LABELHEIGHT= option
MACONTROL procedure, 1876

SHEWHART procedure, 1876
LABOTHER= option
PARETO procedure, 986, 1020
LAST= option
PARETO procedure, 986, 1020
LBOXES= option
SHEWHART procedure, 1876
LCLLABEL2= option
SHEWHART procedure, 1877
LCLLABEL= option
MACONTROL procedure, 1877
SHEWHART procedure, 1877
LENDGRID= option
CUSUM procedure, 1877
MACONTROL procedure, 1877
SHEWHART procedure, 1877
LGRID2= option
PARETO procedure, 986, 1020
LGRID= option
CUSUM procedure, 1877
MACONTROL procedure, 1877
PARETO procedure, 986, 1020
SHEWHART procedure, 1877
LHREF= option
CUSUM procedure, 1877
MACONTROL procedure, 1877
PARETO procedure, 986, 1020
SHEWHART procedure, 1877
LIMITN= option
CUSUM procedure, 547
MACONTROL procedure, 787, 842
SHEWHART procedure, 1877, 1981
LIMLABSUBCHAR= option
SHEWHART procedure, 1879
LINEPRINTER option
PARETO procedure, 959
LLIMITS= option
CUSUM procedure, 547
MACONTROL procedure, 1879
SHEWHART procedure, 1879
LMASK= option
CUSUM procedure, 547
LOTHER= option
PARETO procedure, 986, 1020
LOVERLAY2= option
SHEWHART procedure, 1879
LOVERLAY= option
SHEWHART procedure, 1879
LSL= option
SHEWHART procedure, 1879
LSTARCIRCLES= option
CUSUM procedure, 1879
MACONTROL procedure, 1879
SHEWHART procedure, 1879, 1950, 1951, 1956, 1957
LSTARS= option
CUSUM procedure, 1880
MACONTROL procedure, 1880
SHEWHART procedure, 1880
LTESTS= option
SHEWHART procedure, 1880, 1990
LTMARGIN= option
SHEWHART procedure, 1880, 1936
LTMPLOT= option
SHEWHART procedure, 1881
LVREF= option
CUSUM procedure, 1881
MACONTROL procedure, 1881
PARETO procedure, 987, 1021
SHEWHART procedure, 1881
LZONES= option
CUSUM procedure, 1881
MACONTROL procedure, 1881
SHEWHART procedure, 1881

M

MACHART statement,
See also MACONTROL procedure, MACHART statement
examples, advanced, 859
examples, introductory, 822
overview, 821
syntax, 832
MACONTROL procedure, 758
ANNOTATE2= option, 758
ANNOTATE= option, 758
DATA= data set, 758
FORMCHAR= option, 758, 759
GOUT= option, 759
HISTORY= data set, 760
INSET statement, 862
introduction, 753
LIMITS= data set, 760
LINEPRINTER option, 760
overview, 757
syntax, 758
TABLE= data set, 760
MACONTROL procedure, EWMACHART statement
ALLN option, 810
ALPHA= option, 786
ASYMPTOTIC option, 787, 807
CMEANSYMBOL= option, 787
DATA= data set, 799
HISTORY= data set, 769–771, 800, 801
LIMITN= option, 787, 809, 810
LIMITS= data set, 775, 776, 799, 800, 806
MEANCHAR= option, 787
MEANSYMBOL= option, 787, 814
missing values, 805
MU0= option, 787, 805, 806
NMARKERS option, 810
NOREADLIMITS option, 787
OUTHISTORY= data set, 771, 772, 797
OUTLIMITS= data set, 772, 773, 796
OUTTABLE= data set, 773, 774, 797, 798
READALPHA option, 788
READINDEX= option, 788
READLIMITS option, 789

RESET option, 789
SIGMA0= option, 789, 805, 806
SIGMAS= option, 789
SMETHOD= option, 803, 812
TABLE= data set, 774, 801, 802
VREF= option, 814
WEIGHT= option, 767, 777, 789
XSYMBOL= option, 805
MACONTROL procedure, MACHART statement
ALPHA= option, 841
ASYMPTOTIC option, 842
CMEANSYMBOL= option, 842
DATA= data set, 853, 854
HISTORY= data set, 825–827, 855, 856
LIMITN= option, 842
LIMITS= data set, 806, 830, 831, 854, 855, 860, 861
MEANCHAR= option, 842
MEANSYMBOL= option, 842
missing values, 859
MU0= option, 842, 859, 861
NOREADLIMITS option, 842
OUTHISTORY= data set, 827, 828, 852
OUTLIMITS= data set, 828, 829, 851
OUTTABLE= data set, 829, 830, 852, 853
READALPHA option, 843
READINDEX= option, 843
READLIMITS option, 843
SIGMA0= option, 844, 859, 861
SIGMAS= option, 844
SMETHOD= option, 857, 858
SPAN= option, 823, 832, 844
TABLE= data set, 830, 856
XSYMBOL= option, 860
MAXCMPCT= option
PARETO procedure, 987, 1021
MAXNCAT= option
PARETO procedure, 987, 1021
MAXPANELS= option
CUSUM procedure, 1881
MACONTROL procedure, 1881
SHEWHART procedure, 1881
MCHART statement,
See also SHEWHART procedure, MCHART statement
examples, advanced, 1427
examples, introductory, 1392
options summarized by function, 1406–1408, 1410, 1411, 1413, 1414
overview, 1391
syntax, 1405
MEANCHAR= option
MACONTROL procedure, 787, 842
MEANSYMBOL= option
MACONTROL procedure, 787, 842
MEDCENTRAL= option
SHEWHART procedure, 1881
MINPCT= option
PARETO procedure, 988, 1022
MISSBREAK option
CUSUM procedure, 1882
MACONTROL procedure, 1882
SHEWHART procedure, 1882
MISSING option
PARETO procedure, 988, 1022
missing subgroup variable values
SHEWHART procedure, 1882
MISSING1 option
PARETO procedure, 988, 1022
MISSING2 option
PARETO procedure, 989, 1023
MODEL statement, FACTEX procedure,
See FACTEX procedure, MODEL statement
options summarized by function, 602
syntax, 609
MODEL statement, OPTEX procedure,
See OPTEX procedure, MODEL statement
syntax, 904
MRCHART statement,
See also SHEWHART procedure, MRCHART statement
examples, advanced, 1474
examples, introductory, 1438
options summarized by function, 1450, 1454, 1457–1460
overview, 1437
syntax, 1449
MRRESTART
SHEWHART procedure, 1882
MU0= option
CUSUM procedure, 547
MACONTROL procedure, 787, 842
SHEWHART procedure, 1882, 1981, 2027

N

NAME2= option
SHEWHART procedure, 1882
NAME= option
CUSUM procedure, 1882
MACONTROL procedure, 1882
PARETO procedure, 989, 1023
SHEWHART procedure, 1882
NCOLS= option
PARETO procedure, 989, 1023
NDECIMAL2= option
SHEWHART procedure, 1883
NDECIMAL= option
MACONTROL procedure, 1883
SHEWHART procedure, 1883
NEEDLES option
CUSUM procedure, 1883
MACONTROL procedure, 1883
SHEWHART procedure, 1883
NLEGEND= option
PARETO procedure, 989, 1023
NMARKERS option
CUSUM procedure, 1883
MACONTROL procedure, 1883

SHEWHART procedure, 1883
NO3SIGMACHECK option
SHEWHART procedure, 1883
NOARL option
CUSUM procedure, 547
NOBYREF option
CUSUM procedure, 1883
MACONTROL procedure, 1883
SHEWHART procedure, 1883
NOCHART option
CUSUM procedure, 1884
MACONTROL procedure, 1884
PARETO procedure, 990, 1024
SHEWHART procedure, 1884
NOCHART2 option
SHEWHART procedure, 1884
NOCONNECT option
CUSUM procedure, 1884
MACONTROL procedure, 1884
SHEWHART procedure, 1884
NOCTL option
MACONTROL procedure, 1884
SHEWHART procedure, 1884
NOCTL2 option
SHEWHART procedure, 1884
NOCURVE option
PARETO procedure, 990, 1024
NOFRAME option
CUSUM procedure, 1884
MACONTROL procedure, 1884
PARETO procedure, 990, 1024
SHEWHART procedure, 1884
NOHLABEL option
CUSUM procedure, 1884
MACONTROL procedure, 1884
PARETO procedure, 990, 1024
SHEWHART procedure, 1884
NOHLABEL2 option
PARETO procedure, 1024
NOHLLEG option
PARETO procedure, 990, 1024
NOHTICK option
PARETO procedure, 1024
NOHTICK2 option
PARETO procedure, 1024
NOKEYMOVE option
PARETO procedure, 990, 1024
NOLCL option
MACONTROL procedure, 1884
SHEWHART procedure, 1884
NOLCL2 option
SHEWHART procedure, 1884
NOLEGEND option
CUSUM procedure, 1885
MACONTROL procedure, 1885
SHEWHART procedure, 1885, 1934, 1936, 1937, 2010, 2012
NOLIMIT0 option
SHEWHART procedure, 1885
NOLIMIT1 option
SHEWHART procedure, 1885
NOLIMITLABEL option
MACONTROL procedure, 1885
SHEWHART procedure, 1885
NOLIMITS option
MACONTROL procedure, 1885
SHEWHART procedure, 1885, 2010
NOLIMITSFRAME option
SHEWHART procedure, 1885
NOLIMITSLEGEND option
MACONTROL procedure, 1885
SHEWHART procedure, 1885
NOMASK option
CUSUM procedure, 547
NOOVERLAYLEGEND option
SHEWHART procedure, 1885
NOPHASEFRAME option
SHEWHART procedure, 1885
NOREADLIMITS option
CUSUM procedure, 547
MACONTROL procedure, 787, 842
SHEWHART procedure, 1885
NOTCHES option
SHEWHART procedure, 1886
NOTICKREP option
SHEWHART procedure, 1887
NOTRENDCONNECT option
CUSUM procedure, 1887
MACONTROL procedure, 1887
SHEWHART procedure, 1887
NOTRUNC option
SHEWHART procedure, 1887
NOUCL option
MACONTROL procedure, 1887
SHEWHART procedure, 1887
NOUCL2 option
SHEWHART procedure, 1887
NOVANGLE option
CUSUM procedure, 1888
MACONTROL procedure, 1888
SHEWHART procedure, 1888
NOVLABEL option
PARETO procedure, 990, 1024
NOVLABEL2 option
PARETO procedure, 990
NOVTICK option
PARETO procedure, 990
NOVTICK2 option
PARETO procedure, 990
NPANELPOS= option
CUSUM procedure, 1888
MACONTROL procedure, 1888
SHEWHART procedure, 1888
NPCHART statement, SHEWHART procedure,
See also SHEWHART procedure, NPCHART statement
examples, advanced, 1515
examples, introductory, 1484

options summarized by function, 1495, 1497, 1499, 1500, 1502, 1503
overview, 1483
syntax, 1494
NPSYMBOL= option
SHEWHART procedure, 1888
NROWS= option
PARETO procedure, 990, 1024

O

OPTEX procedure, 889
getting started, 877
learning about OPTEX, 876
order of statements, 889, 895, 905, 922
overview, 875
summary of functions, 890
syntax, 889
OPTEX procedure, BLOCKS statement
COVAR= option, 894, 925
DESIGN= option, 894, 922
INIT= option, 894
ITER= option, 894
KEEP= option, 894
NOEXCHANGE option, 894
options summarized by function, 890
STRUCTURE= option, 894, 920
VAR= option, 925
OPTEX procedure, CLASS statement
DESCENDING option, 895
example, 878
ORDER= option, 895
PARAM= option, 896
REF= option, 899
syntax, 895
TRUNCATE option, 899
OPTEX procedure, EXAMINE statement
DESIGN option, 900
INFORMATION option, 900
NUMBER= option, 900
options summarized by function, 891
VARIANCE option, 900
OPTEX procedure, GENERATE statement
AUGMENT= option, 901, 914
CRITERION= option, 901, 929
INITDESIGN= option, 902, 913
ITER= option, 903
KEEP= option, 903
METHOD= option, 903, 910
N= option, 882, 904, 913
options summarized by function, 890, 891
OPTEX procedure, ID statement, 904
OPTEX procedure, MODEL statement
example, 878
NOINT option, 905, 928
options summarized by function, 890
PRIOR= option, 905, 917
OPTEX procedure, OUTPUT statement
BLOCKNAME= option, 906
NUMBER= option, 906, 908
options summarized by function, 891
OUT= option, 906
OPTEX procedure, PROC OPTEX statement
CODING= option, 892, 922
DATA= option, 892
EPSILON= option, 892
example, 878
NAMELEN option, 892
NOCODE option, 892, 893, 928
NOPRINT option, 893
options summarized by function, 890, 891
SEED= option, 893
STATUS= option, 893
OPTTEX procedure, PROC OPTTEX statement
ODS tables, 948
ORDER1= option
PARETO procedure, 990, 1024
ORDER2= option
PARETO procedure, 991, 1025
ORDER= option
CLASS statement (OPTEX), 895
ORIGIN= option
CUSUM procedure, 547
OTHER= option
PARETO procedure, 991, 1025
OTHERCVAL= option
PARETO procedure, 992, 1026
OTHERNVAL= option
PARETO procedure, 992, 1026
OUT= data set
PARETO procedure, 992, 1026
OUTBOX= option
SHEWHART procedure, 1889
OUTHIGHHTML= option
SHEWHART procedure, 1889
OUTHISTORY= option
CUSUM procedure, 1889
MACONTROL procedure, 1889
SHEWHART procedure, 1889
OUTINDEX= option
CUSUM procedure, 1889
MACONTROL procedure, 1889
SHEWHART procedure, 1889
OUTLABEL2= option
SHEWHART procedure, 1890
OUTLABEL= option
CUSUM procedure, 1889
MACONTROL procedure, 1889
SHEWHART procedure, 1889
OUTLIMITS= option
CUSUM procedure, 1890
MACONTROL procedure, 1890
SHEWHART procedure, 1890
OUTLOWHTML= option
SHEWHART procedure, 1890
OUTPHASE= option
CUSUM procedure, 1890
MACONTROL procedure, 1890
SHEWHART procedure, 1890

OUTPUT statement, CAPABILITY procedure,
See CAPABILITY procedure, OUTPUT statement
getting started, 393
keywords summarized by function, 398, 400
overview, 393
syntax, 396
OUTPUT statement, FACTEX procedure,
See FACTEX procedure, OUTPUT statement
options summarized by function, 602, 603
syntax, 611
OUTPUT statement, OPTEX procedure,
See OPTEX procedure, OUTPUT statement
syntax, 905
OUTTABLE= option
CUSUM procedure, 1891
MACONTROL procedure, 1891
SHEWHART procedure, 1891
OVERLAY2= option
SHEWHART procedure, 1891
OVERLAY2HTML= option
SHEWHART procedure, 1891
OVERLAY2ID= option
SHEWHART procedure, 1891
OVERLAY2SYM= option
SHEWHART procedure, 1891
OVERLAY2SYMHT= option
SHEWHART procedure, 1892
OVERLAY= option
SHEWHART procedure, 1891
OVERLAYCLIPSYM= option
SHEWHART procedure, 1892
OVERLAYCLIPSYMHT= option
SHEWHART procedure, 1892
OVERLAYHTML= option
SHEWHART procedure, 1892
OVERLAYID= option
SHEWHART procedure, 1892
OVERLAYLEGLAB= option
SHEWHART procedure, 1892
OVERLAYSYM= option
SHEWHART procedure, 1892
OVERLAYSYMHT= option
SHEWHART procedure, 1892

P

P0= option
SHEWHART procedure, 1892
PAGENUM= option
CUSUM procedure, 1893
MACONTROL procedure, 1893
SHEWHART procedure, 1893
PAGENUMPOS= option
CUSUM procedure, 1893
MACONTROL procedure, 1893
SHEWHART procedure, 1893
PARAM= option
CLASS statement (OPTEX), 896
PARETO procedure, 958
examples, advanced, 1056
examples, introductory, 963, 999
options summarized by function, 958
overview, 953
syntax, 958
PARETO procedure, BY statement, 1056, 1057
PARETO procedure, HBAR statement
ANCHOR= option, 1010
ANGLE= option, 1011
ANNOKEY option, 1011
ANNOTATE2= data set, 1011
ANNOTATE= data set, 1011
AXISFACTOR= option, 1011
BARLABEL= option, 1012
BARLABPOS= option, 1012
BARLEGEND= option, 1012
BARLEGLABEL= option, 1013
BARWIDTH= option, 1013
CATLEGLABEL= option, 1013
CAXIS2= option, 1013
CAXIS= option, 1013
CBARLINE= option, 1013
CBARS= option, 1014
CCONNECT= option, 1014
CFRAME= option, 1014
CFRAMENLEG= option, 1002, 1003, 1014
CFRAMESIDE= option, 1014
CFRAMETOP= option, 1015
CGRID2= option, 1015
CGRID= option, 1015
CHIGH(n)= option, 1015
CHREF= option, 1015
CLASS= option, 1015
CLASSKEY= option, 1016
CLOW(n)= option, 1016
CMPCTLABEL option, 1012, 1016
COTHER= option, 1016
CPROP= option, 1017
CTEXT= option, 1017
CTEXTSIDE= option, 1017
CTEXTTOP= option, 1017
CTILES= option, 1017
CVREF= option, 1017
DESCRIPTION= option, 1018
FONT= option, 1018
FREQ= option, 1002, 1003, 1018
FRONTREF option, 1018
GRID option, 1018
GRID2 option, 1018
HAXIS2= option, 1018
HAXIS2LABEL= option, 1019
HAXIS= option, 1018
HAXISLABEL= option, 1018
HEIGHT= option, 1019
HLLEGLABEL= option, 1019
HOFFSET= option, 1019
HREF2= option, 1019
HREF2LABELS= option, 1019
HREF= option, 1019

HREFLABELS= option, 1019
HREFLABPOS= option, 1019
INFONT= option, 1020
INHEIGHT= option, 1020
INTERBAR= option, 1002, 1020
INTERTILE= option, 1020
LABOTHER= option, 1020
LAST= option, 1002, 1003, 1020
LGRID2= option, 1020
LGRID= option, 1020
LHREF= option, 1020
LOTHER= option, 1020
LVREF= option, 1021
MAXCMPCT= option, 1021
MAXNCAT= option, 1003, 1005, 1021
MINPCT= option, 1022
MISSING option, 1022
MISSING1 option, 1022
MISSING2 option, 1023
NAME= option, 1023
NCOLS= option, 1023
NLEGEND option, 1023
NLEGEND= option, 1002, 1003, 1023
NOCHART option, 1024
NOCURVE option, 1024
NOFRAME option, 1024
NOHLABEL option, 1024
NOHLABEL2 option, 1024
NOHTICK option, 1024
NOHTICK2 option, 1024
NOKEYMOVE option, 1024
NOVLABEL option, 1024
NROWS= option, 1024
options summarized by function, 1005–1010
ORDER1= option, 1024
ORDER2= option, 1025
OTHER= option, 1003, 1005, 1021, 1022, 1025
OTHERCVAL= option, 1026
OTHERNVAL= option, 1026
OUT= option, 1026
PBARS= option, 1026
PHIGH(n)= option, 1027
PLOW(n)= option, 1027
POTHER= option, 1027
SCALE= option, 1002, 1003, 1027
syntax, 1005
TILELEGEND= option, 1027
TILELEGLABEL= option, 1028
VOFFSET= option, 1028
VREF= option, 1028
VREFLABELS= option, 1028
VREFLABPOS= option, 1028
WAXIS= option, 1028
WBARLINE= option, 1028
WEIGHT= option, 1028
WGRID2= option, 1029
WGRID= option, 1028
PARETO procedure, INSET statement
CFILL= option, 1041
CFILLH= option, 1041
CFRAME= option, 1041
CHEADER= option, 1041
CSHADOW= option, 1041
CTEXT= option, 1042
DATA option, 1042
FONT= option, 1042
FORMAT= option, 1042
HEADER= option, 1042
HEIGHT= option, 1042
NOFRAME option, 1042
POSITION= option, 1042–1044
REFPOINT= option, 1042
PARETO procedure, PROC PARETO statement
ANNOTATE2= data set, 959
ANNOTATE= data set, 959
DATA= data set, 959
FORMCHAR= option, 959
GOUT= option, 959
LINEPRINTER option, 959
PARETO procedure, VBAR statement
ANCHOR= option, 977, 1052, 1061, 1062
ANGLE= option, 977
ANNOKEY option, 977
ANNOTATE2= data set, 977
ANNOTATE= data set, 977
AXISFACTOR= option, 978, 1052
BARLABEL= option, 978
BARLABPOS= option, 978
BARLEGEND= option, 978, 1068, 1069
BARLEGLABEL= option, 979
BARWIDTH= option, 979
CATLEGLABEL= option, 979, 1062, 1063
CAXIS2= option, 980
CAXIS= option, 979
CBARLINE= option, 980
CBARS= option, 980, 1061, 1062, 1068, 1069
CCONNECT= option, 980
CFRAME= option, 980
CFRAMENLEG= option, 966, 967, 981, 1059
CFRAMESIDE= option, 981
CFRAMETOP= option, 981
CGRID2= option, 981
CGRID= option, 981
CHIGH(n)= option, 981, 1066, 1067
CHREF= option, 981
CLASS= option, 981, 1055, 1058, 1062, 1063, 1065, 1066
CLASSKEY= option, 982, 1058
CLOW(n)= option, 983, 1066
CMPCTLABEL option, 978, 983
CONNECTCHAR= option, 983
COTHER= option, 983
CPROP= option, 983, 1059
CTEXT= option, 983
CTEXTSIDE= option, 983
CTEXTTOP= option, 983
CTILES= option, 984, 1069, 1070
CVREF= option, 984

DESCRIPTION= option, 984
FONT= option, 984
FREQ= option, 966, 967, 984
FRONTREF option, 984
GRID option, 984
GRID2 option, 985
HEIGHT= option, 985
HLLEGLABEL= option, 985
HOFFSET= option, 985
HREF= option, 985
HREFCHAR= option, 985
HREFLABELS= option, 985
HREFLABPOS= option, 985
INFONT= option, 986
INHEIGHT= option, 986
INTERBAR= option, 966, 967, 986
INTERTILE= option, 986, 1059
LABOTHER= option, 986
LAST= option, 966, 967, 986
LGRID2= option, 986
LGRID= option, 986
LHREF= option, 986
LOTHER= option, 986
LVREF= option, 987, 1064
MAXCMPCT= option, 987
MAXNCAT= option, 968, 970, 987
MINPCT= option, 988
MISSING option, 988, 1055
MISSING1 option, 988, 1055
MISSING2 option, 989, 1055
NAME= option, 989
NCOLS= option, 989, 1051, 1064–1066
NLEGEND option, 989, 1061, 1062
NLEGEND= option, 966, 967, 989, 1059
NOCHART option, 990
NOCURVE option, 990, 1054, 1062, 1063
NOFRAME option, 990
NOHLABEL option, 990, 1062, 1063
NOKEYMOVE option, 990
NOVLABEL option, 990
NOVLABEL2 option, 990
NOVTICK option, 990
NOVTICK2 option, 990
NROWS= option, 990, 1051, 1064–1066
options summarized by function, 971–976
ORDER1= option, 990, 1055
ORDER2= option, 991, 1055
OTHER= option, 968, 970, 987, 988, 991
OTHERCVAL= option, 992, 1054
OTHERNVAL= option, 992, 1054
OUT= data set, 1053, 1054
OUT= option, 992
PBARS= option, 992, 1061, 1062, 1068, 1069
PHIGH(n)= option, 993, 1066, 1067
PLOW(n)= option, 993, 1066
POTHER= option, 993
SCALE= option, 966, 967, 993, 1054, 1059
SYMBOLCHAR= option, 993
syntax, 970
TILELEGEND= option, 993, 1069, 1070
TILELEGLABEL= option, 994
TURNVLABEL option, 994
VAXIS2= option, 994
VAXIS2LABEL= option, 994
VAXIS= option, 994
VAXISLABEL= option, 994
VOFFSET= option, 994
VREF2= option, 994
VREF2LABELS= option, 995
VREF= option, 994, 1064
VREFCHAR= option, 995
VREFLABELS= option, 995
VREFLABPOS= option, 995
WAXIS= option, 995
WBARLINE= option, 995
WEIGHT= option, 995, 1075, 1076
WGRID2= option, 995
WGRID= option, 995
PATTERN statement, 332
PBARS= option
PARETO procedure, 992, 1026
PCHART statement, ANOM procedure,
See also ANOM procedure, PCHART statement
examples, advanced, 81
examples, introductory, 55
options summarized by function, 64–67, 69, 70
overview, 55
syntax, 63
PCHART statement, SHEWHART procedure,
See also SHEWHART procedure, PCHART statement
examples, advanced, 1558
examples, introductory, 1528
options summarized by function, 1539–1541, 1543–1545, 1547
overview, 1527
syntax, 1537
PCTLDEF= option
SHEWHART procedure, 1893
_PHASE_variables
SHEWHART procedure, 1936
PHASEBREAK option
CUSUM procedure, 1893
MACONTROL procedure, 1893
SHEWHART procedure, 1893, 1987, 1989, 2023
PHASELABTYPE= option
CUSUM procedure, 1893
MACONTROL procedure, 1893
SHEWHART procedure, 1893
PHASELEGEND option
CUSUM procedure, 1894
MACONTROL procedure, 1894
SHEWHART procedure, 1894, 1937, 1941–1946
PHASELIMITS option
CUSUM procedure, 1894
MACONTROL procedure, 1894

SHEWHART procedure, 1894
PHASEMEANSYMBOL= option
SHEWHART procedure, 1298, 1894
PHASEREF option
CUSUM procedure, 1894
MACONTROL procedure, 1894
SHEWHART procedure, 1894, 1937, 1941–1946
phases of subgroups
SHEWHART procedure, 1866
PHASEVALSEP option
CUSUM procedure, 1894
MACONTROL procedure, 1894
SHEWHART procedure, 1894
PHASEVARLABEL option
CUSUM procedure, 1894
MACONTROL procedure, 1894
SHEWHART procedure, 1894
PHIGH(n)= option
PARETO procedure, 993, 1027
PLOW(n)= option
PARETO procedure, 993, 1027
POINTSHTML= option
SHEWHART procedure, 1894
POTHER= option
PARETO procedure, 993, 1027
PPPLOT statement,
See CAPABILITY procedure, PPPLOT statement
getting started, 410
options dictionary, 415
options summarized by function, 412–414
overview, 409
syntax, 411
PROBACC2 function, 2104, 2105, 2115
PROBBNML function, 2106, 2107, 2113, 2114
PROBHYPR function, 2107, 2109, 2110, 2113
PROBMED function, 2110, 2111
PROBPLOT statement,
See CAPABILITY procedure, PROBPLOT statement
getting started, 431
options summarized by function, 438–440
overview, 431
syntax, 437
PROC CAPABILITY statement,
See CAPABILITY procedure, PROC CAPABILITY statement
examples, 218
getting started, 166
options summarized by function, 173
overview, 165
syntax, 173
PROC FACTEX statement,
See FACTEX procedure, PROC FACTEX statement
options summarized by function, 602
syntax, 605
PROC OPTEX statement,
See OPTEX procedure, PROC OPTEX statement
syntax, 892
PROC SHEWHART statement
options summarized by function, 1232
processes, CUSUM procedure
XCHART statement, 535
processes, MACONTROL procedure
EWMACHART statement, 777
MACHART statement, 832
processes, SHEWHART procedure
BOXCHART statement, 1252
CCHART statement, 1315
IRCHART statement, 1357
MCHART statement, 1405
MRCHART statement, 1449
NPCHART statement, 1494
PCHART statement, 1537
RCHART statement, 1585
SCHART statement, 1624
UCHART statement, 1662
XCHART statement, 1703
XRCHART statement, 1750
XSCHART statement, 1801
PSYMBOL= option
SHEWHART procedure, 1894

Q

QC, 1
QQPLOT statement,
See CAPABILITY procedure, QQPLOT statement
getting started, 463
options summarized by function, 468, 470, 471
overview, 463
syntax, 467

R

RANGES option
SHEWHART procedure, 1895
RCHART statement,
See also SHEWHART procedure, RCHART statement
examples, advanced, 1605
examples, introductory, 1574
options summarized by function, 1586–1590, 1592, 1593
overview, 1573
syntax, 1585
READALPHA option
MACONTROL procedure, 788, 843
SHEWHART procedure, 1895
READINDEX= option
CUSUM procedure, 548
MACONTROL procedure, 788, 843
SHEWHART procedure, 1895, 1939–1947, 1987, 1989
READLIMITS option
CUSUM procedure, 548

MACONTROL procedure, 789, 843
SHEWHART procedure, 1896
READPHASES= option
CUSUM procedure, 1897
MACONTROL procedure, 1897
SHEWHART procedure, 1897, 1936–1947, 1986, 1987, 1989
READSIGMAS option
CUSUM procedure, 548
REF= option
CLASS statement (OPTEX), 899
RELIABILITY procedure, 1132
details, 1174
examples, 1085
overview, 1083
syntax, 1132
RELIABILITY procedure, ANALYZE statement, 1129, 1134, 1135, 1138
CONVERGE= option, 1189
PPOS= option, 1178–1180
PREDICT option, 1129
summary of options, 1135, 1138
TOLERANCE option, 1129
RELIABILITY procedure, BY statement, 1132, 1133
RELIABILITY procedure, CLASS statement, 1132, 1133, 1138
RELIABILITY procedure, DISTRIBUTION statement, 1085, 1086, 1093, 1094, 1097, 1098, 1105, 1129, 1132, 1133, 1138
RELIABILITY procedure, FMODE statement, 1132, 1133, 1139
RELIABILITY procedure, FREQ statement, 1093, 1094, 1097, 1098, 1132, 1133, 1140
RELIABILITY procedure, INSET statement, 1132, 1133, 1141–1143
keywords, 1141, 1142
summary of options, 1143
RELIABILITY procedure, MAKE statement, 1105, 1132, 1133, 1144
NOPRINT option, 1144
table keywords, 1144
RELIABILITY procedure, MCFPLOT statement, 1120, 1121, 1123, 1132, 1145, 1146, 1148, 1149, 1152
MCFDIFF option, 1123
summary of options, 1146, 1148, 1149, 1152
RELIABILITY procedure, MODEL statement, 1093, 1094, 1105, 1132, 1152–1155
CONVERGE= option, 1189
CORRB option, 1105
COVB option, 1105
No intercept option, 1190
OBSTATS option, 1093, 1094, 1105, 1155
RELATION= option, 1093, 1094
summary of options, 1154, 1155
RELIABILITY procedure, NENTER statement, 1097, 1098, 1132, 1133, 1156
RELIABILITY procedure, ODS
table keywords, 1214
RELIABILITY procedure, ODS table names, 1214
RELIABILITY procedure, PPLOT statement,
see RELIABILITY procedure, PROBPLOT statement
RELIABILITY procedure, PROBPLOT statement, 1085, 1086, 1089, 1097, 1098, 1132, 1156, 1158, 1160–1162, 1164
CONVERGE= option, 1189
COVB option, 1085, 1086
NOCONF option, 1089, 1097, 1098
OVERLAY option, 1089
PCONFPLT option, 1097, 1098
PPOS= option, 1178–1180
READOUT option, 1097, 1098
summary of options, 1158, 1160–1162, 1164
RELIABILITY procedure, RELATIONPLOT statement, 1093, 1094, 1132, 1164, 1166, 1167, 1169, 1171–1173
CONVERGE= option, 1189
FIT= option, 1093, 1094
LUPPER= option, 1093, 1094
NOCONF option, 1093, 1094
PLOTDATA option, 1093, 1094
PLOTFIT option, 1093, 1094
PPLOT option, 1093, 1094
PPOS= option, 1178–1180
RELATION= option, 1093, 1094
SLOWER= option, 1093, 1094
summary of options, 1166, 1167, 1169, 1171–1173
RELIABILITY procedure, RPLOT statement,
see RELIABILITY procedure, RELATIONPLOT statement
RELIABILITY procedure, UNITID statement, 1120, 1123, 1132, 1133, 1173
REPEAT option
CUSUM procedure, 1899
MACONTROL procedure, 1899
SHEWHART procedure, 1899
RESET option
MACONTROL procedure, 789
responses, ANOM procedure
BOXCHART statement, 120
PCHART statement, 63
UCHART statement, 90
XCHART statement, 23
RSYMBOL= option
SHEWHART procedure, 1899
RTMARGIN= option
SHEWHART procedure, 1899
RTMPLOT= option
SHEWHART procedure, 1899, 2028

S

SAS/QC, 1
SCALE= option
PARETO procedure, 993, 1027
SCHART statement,

See also SHEWHART procedure, SCHART statement
examples, advanced, 1644
examples, introductory, 1614
options summarized by function, 1625–1630, 1632, 1634
overview, 1613
syntax, 1624
SCHEME= option
CUSUM procedure, 549
SEPARATE option
SHEWHART procedure, 1900
SERIFS option
SHEWHART procedure, 1900
SHEWHART procedure, 1230
and PROC ARIMA, 2003, 2005–2009
and PROC CAPABILITY, 2028, 2031
and PROC MACONTROL, 2007
and PROC MIXED, 2014, 2015
and PROC PRINCOMP, 2034
syntax, 1230
SHEWHART procedure, all chart statements
ALLLABEL= option, 1854
ALPHA= option, 1855
ANNOTATE= option, 1855
BILEVEL option, 1856
BLOCKLABELPOS= option, 1856
BLOCKLABTYPE= option, 1857
BLOCKPOS= option, 1857
BLOCKREP option, 1858
CAXIS= option, 1861
CBLOCKLAB= option, 1861
CBLOCKVAR= option, 1861
CCONNECT= option, 1863
CCOVERLAY2= option, 1863
CCOVERLAY= option, 1863
CFRAME= option, 1863
CFRAMELAB= option, 1863
CGRID= option, 1863
CHREF= option, 1864
CINFILL= option, 1864
CLABEL= option, 1864
CLIMITS= option, 1864
CONNECTCHAR= option, 1866
COUT= option, 1866
COUTFILL= option, 1866
COVERLAY2= option, 1866
COVERLAY= option, 1866
COVERLAYCLIP= option, 1866
CPHASELEG= option, 1867
CTESTLABBOX= option, 1869
CTESTS= option, 1869
CTESTSYMBOL= option, 1869
CTEXT= option, 1869
CVREF= option, 1869
CZONES= option, 1870
DESCRIPTION= option, 1870
ENDGRID option, 1870
EXCHART option, 1870
FONT= option, 1870
GRID option, 1871
HAXIS= option, 1871
HEIGHT= option, 1871
HMINOR= option, 1872
HOFFSET= option, 1872
HREF2DATA= option, 1873
HREF= option, 1872
HREFCHAR= option, 1873
HREFDATA= option, 1873
HREFLABELS= option, 1873
HREFLABPOS= option, 1873
HTML2= option, 1874
HTML= option, 1874
HTML_LEGEND= option, 1874
INTERVAL= option, 1875
INTSTART= option, 1876
LABELANGLE= option, 1876
LABELFONT= option, 1876
LABELHEIGHT= option, 1876
LCLLABEL= option, 1877
LENDGRID= option, 1877
LGRID= option, 1877
LHREF= option, 1877
LIMITN= option, 1877
LLIMITS= option, 1879
LOVERLAY2= option, 1879
LOVERLAY= option, 1879
LTESTS= option, 1880
LVREF= option, 1881
LZONES= option, 1881
MAXPANELS= option, 1881
NAME= option, 1882
NDECIMAL= option, 1883
NO3SIGMACHECK option, 1883
NOBYREF option, 1883
NOCHART option, 1884
NOCONNECT option, 1884
NOCTL option, 1884
NOFRAME option, 1884
NOHLABEL option, 1884
NOLCL option, 1884
NOLEGEND option, 1885
NOLIMITLABEL option, 1885
NOLIMITS option, 1885
NOLIMITSFRAME option, 1885
NOLIMITSLEGEND option, 1885
NOOVERLAYLEGEND option, 1885
NOPHASEFRAME option, 1885
NOREADLIMITS option, 1885
NOUCL option, 1887
NOVANGLE option, 1888
NPANELPOS= option, 1888
OUTHIGHHTML= option, 1889
OUTHISTORY= option, 1889
OUTINDEX= option, 1889
OUTLABEL= option, 1889
OUTLIMITS= option, 1890
OUTLOWHTML= option, 1890

OUTPHASE= option, 1890
OUTTABLE= option, 1891
OVERLAY2= option, 1891
OVERLAY2HTML= option, 1891
OVERLAY2ID= option, 1891
OVERLAY2SYM= option, 1891
OVERLAY2SYMHT= option, 1892
OVERLAY= option, 1891
OVERLAYCLIPSYM= option, 1892
OVERLAYCLIPSYMHT= option, 1892
OVERLAYHTML= option, 1892
OVERLAYID= option, 1892
OVERLAYLEGLAB= option, 1892
OVERLAYSYM= option, 1892
OVERLAYSYMHT= option, 1892
PAGENUM= option, 1893
PAGENUMPOS= option, 1893
PHASEBREAK option, 1893
PHASELABTYPE= option, 1893
PHASELEGEND option, 1894
PHASELIMITS option, 1894
PHASEREF option, 1894
PHASEVALSEP option, 1894
PHASEVARLABEL option, 1894
POINTSHTML= option, 1894
READALPHA option, 1895
READINDEX= option, 1895
READLIMITS option, 1896
READPHASES= option, 1897
REPEAT option, 1899
SIGMAS= option, 1900
SKIPHLABELS= option, 1901
SYMBOLCHARS= option, 1909
SYMBOLLEGEND= option, 1909
SYMBOLORDER= option, 1910
TABLE option, 1910
TABLEALL option, 1910
TABLECENTRAL option, 1910
TABLEID option, 1911
TABLELEGEND option, 1911
TABLEOUTLIM option, 1911
TABLETESTS option, 1911
TEST2RESET= option, 1911
TEST2RUN= option, 1912
TEST3RUN= option, 1912
TESTACROSS option, 1912
TESTCHAR= option, 1912
TESTFONT= option, 1912
TESTHEIGHT= option, 1913
TESTLABBOX option, 1913
TESTLABEL*n*= option, 1914
TESTLABEL= option, 1913
TESTNMETHOD= option, 1914
TESTOVERLAP option, 1914
TESTRESET= option, 1915
TESTS= option, 1915
TOTPANELS= option, 1917
TURNALL option, 1918
TURNHLABELS option, 1918
TYPE= option, 1918
UCLLABEL= option, 1919
VAXIS= option, 1920
VFORMAT2= option, 1920
VFORMAT= option, 1920
VMINOR= option, 1921
VOFFSET= option, 1921
VREF= option, 1921
VREFCHAR= option, 1922
VREFLABELS= option, 1922
VREFLABPOS= option, 1922
WAXIS= option, 1923
WEBOUT= option, 1923
WGRID= option, 1923
WLIMITS= option, 1923
WNEEDLES= option, 1924
WOVERLAY2= option, 1924
WOVERLAY= option, 1924
WTESTS= option, 1924
ZEROSTD= option, 1925
ZONECHAR= option, 1926
ZONEVALPOS= option, 1926
SHEWHART procedure, BOXCHART statement, 1302,
See also SHEWHART procedure, all chart statements
ALPHA= option, 1268
BOX= data set, 1278
BOXSTYLE= option, 1287–1290
BOXWIDTHSCALE= option, 1292, 1293
CONTROLSTAT= option, 1242, 1267, 1268
DATA= data set, 1274, 1275
HISTORY= data set, 1243–1246, 1276, 1277
LBOXES= option, 1293, 1295
LIMITN= option, 1268
LIMITS= data set, 1251, 1252, 1275
LSL= option, 1270
MEDCENTRAL= option, 1268
missing values, 1283
MU0= option, 1268
NOCHART option, 1246
NOHLABEL option, 1830, 1831
NOLEGEND option, 1830, 1831, 2010
NOLIMITS option, 2010
NOTCHES option, 1291
OUTBOX= data set, 1270
OUTBOX= option, 1889
OUTHISTORY= data set, 1246–1248, 1271, 1272
OUTLIMITS= data set, 1248, 1249, 1269, 1270
OUTTABLE= data set, 1249, 1250, 1273, 1274
RANGES option, 1247, 1895
SERIFS option, 1287
SIGMA0= option, 1268
SIGMAS= option, 1268
SMETHOD= option, 1280–1282
STDDEVIATIONS option, 2010
TABLE= data set, 1250, 1277, 1278
TARGET= option, 1270

TESTS= option, 1977
USL= option, 1270
SHEWHART procedure, CCHART statement,
See also SHEWHART procedure, all chart statements
ALPHA= option, 1329
CSYMBOL= option, 1340, 1341
DATA= data set, 1332
HISTORY= data set, 1311–1313, 1333, 1334
LIMITN= option, 1329
LIMITS= data set, 1310, 1311, 1333
LTESTS= option, 1337, 1338, 1340
missing values, 1336
NOCHART option, 1308
NOLEGEND option, 1340, 1341
OUTHISTORY= data set, 1313, 1315, 1330, 1331
OUTLIMITS= data set, 1308, 1329, 1330
OUTTABLE= data set, 1308, 1309, 1331, 1332
SIGMAS= option, 1329
SUBGROUPN= option, 1313, 1315
TABLE= data set, 1309, 1334, 1336
TABLELEGEND option, 1337, 1338, 1340
TABLETESTS option, 1337, 1338, 1340
TESTS= option, 1337, 1338, 1340, 1977
U0= option, 1329, 1340, 1341
ZONELABELS option, 1337, 1338, 1340
SHEWHART procedure, INSET statement
CFILL= option, 1845
CFILLH= option, 1845
CFRAME= option, 1845
CHEADER= option, 1845
CSHADOW= option, 1845
CTEXT= option, 1846
DATA option, 1846
FONT= option, 1846
FORMAT= option, 1846
HEADER= option, 1846
HEIGHT= option, 1846
NOFRAME option, 1846
POSITION= option, 1846–1848
REFPOINT= option, 1846
SHEWHART procedure, IRCHART statement,
See also SHEWHART procedure, all chart statements
ALPHA= option, 1372
DATA= data set, 1376
HISTORY= data set, 1351, 1352, 1377, 1378
LIMITN= option, 1356, 1372
LIMITS= data set, 1354, 1355, 1376, 1377
LSL= option, 1373
LTESTS= option, 1382, 1383
LTMARGIN= option, 1387
LTMPLOT= option, 1387
missing values, 1381
MU0= option, 1372, 1383, 2027
NOCHART option, 1350
OUTHISTORY= data set, 1350, 1374
OUTLIMITS= data set, 1352, 1373, 1374
OUTTABLE= data set, 1353, 1374, 1375
PHASEBREAK option, 2023
RTMPLOT= option, 1386, 1387, 2028
SIGMA0= option, 1372, 1383, 2027
SIGMAS= option, 1372
TABLE= data set, 1354, 1378, 1379
TABLETESTS option, 1382, 1383
TARGET= option, 1374
TEST2RUN= option, 1382, 1383
TESTS= option, 1382, 1383, 1977
USL= option, 1373
XSYMBOL= option, 1383
ZONELABELS option, 1382, 1383
SHEWHART procedure, MCHART statement,
See also SHEWHART procedure, all chart statements
ALPHA= option, 1419
DATA= data set, 1422, 1423
HISTORY= data set, 1394–1396, 1398, 1424, 1425
LIMITN= option, 1419
LIMITS= data set, 1403, 1404, 1423, 1424
LSL= option, 1420
MEDCENTRAL= option, 1419, 1428, 1429
missing values, 1426
MU0= option, 1419, 1429, 1430
NDECIMAL= option, 1427
NOCHART option, 1398, 1399
OUTHISTORY= data set, 1398–1400, 1420, 1421
OUTLIMITS= data set, 1400, 1401, 1419, 1420
OUTTABLE= data set, 1401, 1402, 1421, 1422
SIGMA0= option, 1419
SIGMAS= option, 1419
SMETHOD= option, 1431–1433
STDDEVIATIONS option, 1400, 1432, 1433
TABLE= data set, 1402, 1425, 1426
TARGET= option, 1420
TESTS= option, 1977
USL= option, 1420
XSYMBOL= option, 1429, 1430
SHEWHART procedure, MRCHART statement,
See also SHEWHART procedure, all chart statements
ALLN option, 1476, 1477
ALPHA= option, 1465
DATA= data set, 1469
HISTORY= data set, 1440–1442, 1444, 1470, 1471
LIMITN= option, 1465, 1475–1477
LIMITS= data set, 1447, 1448, 1469, 1470
MEDCENTRAL= option, 1465
missing values, 1473
MU0= option, 1465
NMARKERS option, 1476, 1477
NOCHART option, 1444
OUTHISTORY= data set, 1444, 1445, 1466, 1467
OUTLIMITS= data set, 1445, 1465, 1466

OUTTABLE= data set, 1446, 1447, 1467, 1468
SIGMA0= option, 1465
SIGMAS= option, 1465
SMETHOD= option, 1472, 1473, 1477, 1478
TABLE= data set, 1446, 1471, 1472
TESTS2= option, 1991
TESTS= option, 1977
SHEWHART procedure, NPCHART statement,
See also SHEWHART procedure, all chart statements
ALLN option, 1519, 1520
ALPHA= option, 1508
DATA= data set, 1511
DATAUNIT= option, 1486, 1487
HISTORY= data set, 1487, 1488, 1512, 1513
LIMITN= option, 1508, 1519, 1520
LIMITS= data set, 1492, 1512, 1517, 1518, 1521–1523
LTESTS= option, 1515, 1516
missing values, 1514
NEEDLES option, 1517, 1518
NOLEGEND option, 1517, 1518
NPSYMBOL= option, 1517, 1518
OUTHISTORY= data set, 1488, 1489, 1509
OUTLIMITS= data set, 1489, 1490, 1508, 1518–1520
OUTTABLE= data set, 1490, 1509, 1510
P0= option, 1508, 1517, 1518
SIGMAS= option, 1508
SUBGROUPN= option, 1485, 1518–1520
TABLE= data set, 1490, 1492, 1513, 1514
TABLELEGEND option, 1515, 1516
TABLETESTS option, 1515, 1516
TESTS= option, 1515, 1516, 1977
ZONELABELS option, 1515, 1516
SHEWHART procedure, PCHART statement,
See also SHEWHART procedure, all chart statements
ALLN option, 1564
ALPHA= option, 1551
DATA= data set, 1554
DATAUNIT= option, 1531
FONT= option, 1562
HISTORY= data set, 1531, 1532, 1555, 1556
LIMITN= option, 1551, 1564
LIMITS= data set, 1536, 1555, 1560, 1562
LTESTS= option, 1559
missing values, 1558
NEEDLES option, 1560, 1562
NOLEGEND option, 1560, 1562
OUTHISTORY= data set, 1533, 1552
OUTLIMITS= data set, 1534, 1551, 1562–1564
OUTTABLE= data set, 1535, 1553
P0= option, 1551, 1560, 1562
PSYMBOL= option, 1560, 1562
READINDEX= option, 1566, 1567
SIGMAS= option, 1551
SUBGROUPN= option, 1529, 1562–1564
TABLE= data set, 1535, 1556, 1557
TABLELEGEND option, 1559
TABLETESTS option, 1559
TESTS= option, 1559, 1977
VREF= option, 1566, 1567
VREFLABELS= option, 1566, 1567
VREFLABPOS= option, 1566, 1567
YSCALE= option, 1564
ZONELABELS option, 1559
SHEWHART procedure, PROC SHEWHART statement
CIINDICES= option, 1864
SHEWHART procedure, RCHART statement,
See also SHEWHART procedure, all chart statements
ALPHA= option, 1596, 1605, 1606
DATA= data set, 1600
HISTORY= data set, 1576–1579, 1601, 1602
LIMITN= option, 1596
LIMITS= data set, 1583, 1584, 1601, 1606–1608
LSL= option, 1598
missing values, 1604
NOCHART option, 1579, 1580
NOLIMIT0 option, 1608
OUTHISTORY= data set, 1579, 1580, 1598
OUTLIMITS= data set, 1580, 1581, 1596, 1598, 1605, 1606
OUTTABLE= data set, 1581, 1582, 1599
READALPHA option, 1606
SIGMA0= option, 1596, 1608
SIGMAS= option, 1596
SMETHOD= option, 1603, 1604
TABLE= data set, 1582, 1602, 1603
TARGET= option, 1598
TESTS2= option, 1991
USL= option, 1598
SHEWHART procedure, SCHART statement,
See also SHEWHART procedure, all chart statements
ALPHA= option, 1635
DATA= data set, 1639
HISTORY= data set, 1617–1619, 1640, 1641
LIMITN= option, 1635
LIMITS= data set, 1623, 1624, 1640
LSL= option, 1637
missing values, 1644
OUTHISTORY= data set, 1619, 1620, 1637
OUTLIMITS= data set, 1620, 1621, 1635, 1637
OUTTABLE= data set, 1621, 1622, 1638
SIGMA0= option, 1635, 1644, 1645
SIGMAS= option, 1635
SMETHOD= option, 1642, 1643
SSYMBOL= option, 1644, 1645
TABLE= data set, 1622, 1641, 1642
TARGET= option, 1637
TESTS2= option, 1991
USL= option, 1637
SHEWHART procedure, UCHART statement,

See also SHEWHART procedure, all chart statements
ALPHA= option, 1675
DATA= data set, 1678, 1679
HISTORY= data set, 1657–1659, 1679, 1680
LIMITN= option, 1675
LIMITS= data set, 1656, 1657, 1679
LTESTS= option, 1682, 1683
missing values, 1681
NOCHART option, 1654, 1655
OUTHISTORY= data set, 1660, 1661, 1676, 1677
OUTLIMITS= data set, 1654, 1675, 1676, 1685–1688
OUTTABLE= data set, 1655, 1656, 1677, 1678
SIGMAS= option, 1675
SUBGROUPN= option, 1652, 1653, 1661, 1686–1688
TABLE= data set, 1656, 1680, 1681
TABLETESTS option, 1682, 1683
TESTS= option, 1682, 1683, 1977
U0= option, 1675, 1684, 1685
USYMBOL= option, 1684, 1685
ZONELABELS option, 1682, 1683
SHEWHART procedure, XCHART statement,
See also SHEWHART procedure, all chart statements
ALPHA= option, 1715
BLOCKLABELPOS= option, 1936, 2023, 2024
BLOCKLABTYPE= option, 2023, 2024
BLOCKPOS= option, 1934, 1936
CBLOCKVAR= option, 1934, 1936
CFRAME= option, 1937
CNEEDLES= option, 1961
CPHASELEG= option, 1937
DATA= data set, 1719, 1720
HISTORY= data set, 1694–1697, 1721, 1722
LABELFONT= option, 1956, 1957
LIMITN= option, 1715
LIMITS= data set, 1701–1703, 1720, 1721
LSL= option, 1717
LSTARCIRCLES= option, 1950, 1951, 1956, 1957
LTESTS= option, 1726, 1727
LTMARGIN= option, 1936
missing values, 1726
MU0= option, 1715
NOCHART option, 1697, 1698
NOLEGEND option, 1726, 1727, 1934, 1936, 1937
OUTHISTORY= data set, 1697–1699, 1717, 1718
OUTINDEX= option, 1730
OUTLIMITS= data set, 1699, 1700, 1716, 1717, 1732
OUTTABLE= data set, 1700, 1701, 1719
PHASELEGEND option, 1937, 1941–1946
PHASEREF option, 1937, 1941–1946
READINDEXES= option, 1939–1947
READPHASES= option, 1936–1947
SIGMA0= option, 1715
SIGMAS= option, 1715
SMETHOD= option, 1723–1725, 1729, 1730
STARBDRADIUS= option, 1957
STARCIRCLES= option, 1950, 1951
STARINRADIUS= option, 1951
STARLABEL= option, 1956, 1957
STARLEGEND= option, 1956, 1957
STAROUTRADIUS= option, 1951
STARSPECS= option, 1955–1957
STARSTART= option, 1950–1954, 1956, 1957
STARTYPE= option, 1952–1954
STARVERTICES= option, 1949–1954, 1956, 1957
STDDEVIATIONS option, 1729, 1730
SYMBOLCHARS= option, 1931
SYMBOLLEGEND= option, 1931
TABLE= data set, 1701, 1722, 1723
TABLECENTRAL option, 1726, 1727
TABLELEGEND option, 1726, 1727
TABLETESTS option, 1726, 1727
TARGET= option, 1717
TESTS= option, 1726, 1727, 1977
TRENDVAR= option, 1961, 2023
USL= option, 1717
WSTARCIRCLES= option, 1951
ZONELABELS option, 1726, 1727
SHEWHART procedure, XRCHART statement,
See also SHEWHART procedure, all chart statements
ALLN option, 1783, 1981
ALPHA= option, 1763
CLIPFACTOR= option, 1963, 1965
CLIPLEGEND= option, 1965
CLIPLEGPOS= option, 1965
CLIPSUBCHAR= option, 1965
CLIPSYMBOL= option, 1965
CTESTS= option, 1990
CZONES= option, 1990
DATA= data set, 1767
FONT= option, 1780
HISTORY= data set, 1741–1743, 1768, 1769
LIMITN= option, 1763, 1783, 1981
LIMITS= data set, 1748, 1749, 1768, 1781
LSL= option, 1764
LTESTS= option, 1778, 1990
missing values, 1776
MU0= option, 1763, 1780, 1781, 1981
NMARKERS option, 1784
NOCHART option, 1744, 1745
OUTHISTORY= data set, 1744, 1765
OUTLIMITS= data set, 1745, 1746, 1764, 1765
OUTTABLE= data set, 1746, 1747, 1766
PHASEBREAK option, 1987, 1989
READINDEXES= option, 1987
READPHASES= option, 1986, 1987
SIGMA0= option, 1763, 1780, 1781, 1981
SIGMAS= option, 1763

SMETHOD= option, 1773, 1784, 1984
subgroup-variable, 1771, 1772
TABLE= data set, 1748, 1769, 1770
TABLETESTS option, 1778
TARGET= option, 1765, 1775
TESTACROSS option, 1989, 1990
TESTCHAR= option, 1990
TESTLABEL*n*= option, 1990
TESTLABEL= option, 1985, 1990
TESTNMETHOD= option, 1983, 1989, 1990
TESTS2= option, 1991
TESTS= option, 1977, 1979, 1981
USL= option, 1764
XSYMBOL= option, 1780
ZONECHAR= option, 1990
ZONELABELS option, 1778, 1990
ZONES option, 1990
SHEWHART procedure, XSCHART statement,
See also SHEWHART procedure, all chart statements
ALPHA= option, 1815, 1826
DATA= data set, 1819
HISTORY= data set, 1820, 1821
LIMITN= option, 1815
LIMITS= data set, 1820
LSL= option, 1816
missing values, 1825
MU0= option, 1815
OUTHISTORY= data set, 1796, 1817
OUTLIMITS= data set, 1797, 1816, 1817, 1826
OUTTABLE= data set, 1797, 1798, 1818
SIGMA0= option, 1815
SIGMAS= option, 1815
SPLIT= option, 1969
TABLE= data set, 1798, 1821, 1822
TARGET= option, 1817
TESTS2= option, 1991
TESTS= option, 1977
USL= option, 1816
SHIFT= option
CUSUM procedure, 549
SIGMA0= option
CUSUM procedure, 549
MACONTROL procedure, 789, 844
SHEWHART procedure, 1900, 1981, 2027
SIGMAS= option
CUSUM procedure, 549
MACONTROL procedure, 789, 844
SHEWHART procedure, 1900
SIZE statement, FACTEX procedure,
See FACTEX procedure, SIZE statement
options summarized by function, 601
syntax, 615
SKIPHLABELS= option
CUSUM procedure, 1901
MACONTROL procedure, 1901
SHEWHART procedure, 1901
SMETHOD= option
CUSUM procedure, 549
MACONTROL procedure, 1902
SHEWHART procedure, 1902, 1984
SPAN= option
MACONTROL procedure, 844
SPEC statement
options summarized by function, 184
syntax, 183
SPLIT= option
CUSUM procedure, 1902
MACONTROL procedure, 1902
SHEWHART procedure, 1902, 1969
SSYMBOL= option
SHEWHART procedure, 1902
STARBDRADIUS= option
CUSUM procedure, 1903
MACONTROL procedure, 1903
SHEWHART procedure, 1903, 1957
STARCIRCLES= option
CUSUM procedure, 1903
MACONTROL procedure, 1903
SHEWHART procedure, 1903, 1950, 1951
STARINRADIUS= option
CUSUM procedure, 1904
MACONTROL procedure, 1904
SHEWHART procedure, 1904, 1951
STARLABEL= option
CUSUM procedure, 1905
MACONTROL procedure, 1905
SHEWHART procedure, 1905, 1956, 1957
STARLEGEND= option
CUSUM procedure, 1905
MACONTROL procedure, 1905
SHEWHART procedure, 1905, 1956, 1957
STARLEGENDLAB= option
CUSUM procedure, 1906
MACONTROL procedure, 1906
SHEWHART procedure, 1906
STAROUTRADIUS= option
CUSUM procedure, 1906
MACONTROL procedure, 1906
SHEWHART procedure, 1906, 1951
STARSPECS= option
CUSUM procedure, 1906
MACONTROL procedure, 1906
SHEWHART procedure, 1906, 1955–1957
STARSTART= option
CUSUM procedure, 1907
MACONTROL procedure, 1907
SHEWHART procedure, 1907, 1950–1954, 1956, 1957
STARTYPE= option
CUSUM procedure, 1908
MACONTROL procedure, 1908
SHEWHART procedure, 1908, 1952–1954
STARVERTICES= option
CUSUM procedure, 1908
MACONTROL procedure, 1908
SHEWHART procedure, 1908, 1949–1954, 1956, 1957

STDDEVIATIONS option
SHEWHART procedure, 1902, 1908, 2010
STDMED function, 2111, 2112
subgroup-variable, CUSUM procedure
XCHART statement, 536
subgroup-variable, MACONTROL procedure
EWMACHART statement, 777
MACHART statement, 833
subgroup-variable, SHEWHART procedure
BOXCHART statement, 1252
CCHART statement, 1316
IRCHART statement, 1357
MCHART statement, 1405
MRCHART statement, 1449
NPCHART statement, 1494
PCHART statement, 1538
RCHART statement, 1585
SCHART statement, 1624
UCHART statement, 1662
XCHART statement, 1704
XRCHART statement, 1750
XSCHART statement, 1801
SUBGROUPN= option
SHEWHART procedure, 1909
SYMBOL statement, 332, 333
symbol-variable, ANOM procedure
BOXCHART statement, 121
PCHART statement, 64
UCHART statement, 91
XCHART statement, 24
symbol-variable, CUSUM procedure
XCHART statement, 536
symbol-variable, MACONTROL procedure
EWMACHART statement, 778
MACHART statement, 833
symbol-variable, SHEWHART procedure
BOXCHART statement, 1253
CCHART statement, 1316
displaying, 1909
IRCHART statement, 1358
MCHART statement, 1406
MRCHART statement, 1450
NPCHART statement, 1495
PCHART statement, 1538
RCHART statement, 1586
SCHART statement, 1625
UCHART statement, 1663
XCHART statement, 1704
XRCHART statement, 1751
XSCHART statement, 1802
SYMBOLCHAR= option
PARETO procedure, 993
SYMBOLCHARS= option
CUSUM procedure, 1909
MACONTROL procedure, 1909
SHEWHART procedure, 1909, 1931
SYMBOLLEGEND= option
CUSUM procedure, 1909
MACONTROL procedure, 1909
SHEWHART procedure, 1909, 1931
SYMBOLORDER= option
CUSUM procedure, 1910
MACONTROL procedure, 1910
SHEWHART procedure, 1910

T

TABLE option
MACONTROL procedure, 1910
SHEWHART procedure, 1910
TABLEALL option
CUSUM procedure, 550
MACONTROL procedure, 1910
SHEWHART procedure, 1910
TABLEBOX= option
SHEWHART procedure, 1910
TABLECENTRAL option
MACONTROL procedure, 1910
SHEWHART procedure, 1910
TABLECHART option
CUSUM procedure, 550
TABLECOMP option
CUSUM procedure, 550
TABLEID option
CUSUM procedure, 550
MACONTROL procedure, 1911
SHEWHART procedure, 1911
TABLELEGEND option
SHEWHART procedure, 1911
TABLEOUT option
CUSUM procedure, 550
TABLEOUTLIM option
MACONTROL procedure, 1911
SHEWHART procedure, 1911
tables
extreme observations, number, 180
extreme values, number, 180
robust estimates of scale, 181
specialized capability indices, 182
TABLESUMMARY option
CUSUM procedure, 550
TABLETESTS option
SHEWHART procedure, 1911
TARGET= option
SHEWHART procedure, 1911
TEST2RESET= option
SHEWHART procedure, 1911
TEST2RUN= option
SHEWHART procedure, 1912, 1983
TEST3RUN= option
SHEWHART procedure, 1912, 1983
TESTACROSS option
SHEWHART procedure, 1912, 1989, 1990
TESTCHAR= option
SHEWHART procedure, 1912, 1990
TESTFONT= option
SHEWHART procedure, 1912
TESTHEIGHT= option
SHEWHART procedure, 1913

TESTLABBOX option
SHEWHART procedure, 1913
TESTLABELn= option
SHEWHART procedure, 1914, 1990
TESTLABEL= option
SHEWHART procedure, 1913, 1985, 1990
TESTNMETHOD= option
SHEWHART procedure, 1914, 1983, 1989, 1990
TESTOVERLAP option
SHEWHART procedure, 1914
TESTRESET= option
SHEWHART procedure, 1915
TESTS2= option
SHEWHART procedure, 1917, 1991
TESTS= option
SHEWHART procedure, 1915, 1979, 1981
TESTSYMBOL= option
SHEWHART procedure, 1917
TESTSYMBOLHT= option
SHEWHART procedure, 1917
TILELEGEND= option
PARETO procedure, 993, 1027
TILELEGLABEL= option
PARETO procedure, 994, 1028
TOTPANELS= option
CUSUM procedure, 1917
MACONTROL procedure, 1917
SHEWHART procedure, 1917
TRENDVAR= option
CUSUM procedure, 1917
MACONTROL procedure, 1917
SHEWHART procedure, 1917, 1961, 2023
TRUNCATE option
CLASS statement (OPTEX), 899
TURNALL option
CUSUM procedure, 1918
MACONTROL procedure, 1918
SHEWHART procedure, 1918
TURNHLABELS option
CUSUM procedure, 1918
MACONTROL procedure, 1918
SHEWHART procedure, 1918
TURNVLABEL option
PARETO procedure, 994
TYPE= option
CUSUM procedure, 550
MACONTROL procedure, 1918
SHEWHART procedure, 1918

U

U0= option
SHEWHART procedure, 1919
UCHART statement, ANOM procedure,
See also ANOM procedure, UCHART statement
examples, advanced, 107
examples, introductory, 85
options summarized by function, 91, 92, 94, 96, 97
overview, 85
syntax, 90
UCHART statement, SHEWHART procedure,
See also SHEWHART procedure, UCHART statement
examples, advanced, 1682
examples, introductory, 1652
options summarized by function, 1664, 1666, 1668–1670, 1672
overview, 1651
syntax, 1662
UCLLABEL2= option
SHEWHART procedure, 1919
UCLLABEL= option
MACONTROL procedure, 1919
SHEWHART procedure, 1919
USL= option
SHEWHART procedure, 1919
USYMBOL= option
SHEWHART procedure, 1919

V

VAXIS2= option
PARETO procedure, 994
SHEWHART procedure, 1920
VAXIS2LABEL= option
PARETO procedure, 994
VAXIS= option
CUSUM procedure, 1920
MACONTROL procedure, 1920
PARETO procedure, 994
SHEWHART procedure, 1920
VAXISLABEL= option
PARETO procedure, 994
VFORMAT2= option
SHEWHART procedure, 1920
VFORMAT= option
SHEWHART procedure, 1920
VMINOR= option
CUSUM procedure, 1921
MACONTROL procedure, 1921
SHEWHART procedure, 1921
VOFFSET= option
CUSUM procedure, 1921
MACONTROL procedure, 1921
PARETO procedure, 994, 1028
SHEWHART procedure, 1921
VREF2= option
CUSUM procedure, 1922
MACONTROL procedure, 1922
PARETO procedure, 994
SHEWHART procedure, 1922
VREF2LABELS= option
CUSUM procedure, 1922
MACONTROL procedure, 1922
PARETO procedure, 995
SHEWHART procedure, 1922
VREF= option
CUSUM procedure, 1921

MACONTROL procedure, 1921
PARETO procedure, 994, 1028
SHEWHART procedure, 1921
VREFCHAR= option
CUSUM procedure, 1922
MACONTROL procedure, 1922
PARETO procedure, 995
SHEWHART procedure, 1922
VREFLABELS= option
CUSUM procedure, 1922
MACONTROL procedure, 1922
PARETO procedure, 995, 1028
SHEWHART procedure, 1922
VREFLABPOS= option
CUSUM procedure, 1922
MACONTROL procedure, 1922
PARETO procedure, 995, 1028
SHEWHART procedure, 1922
VZERO option
SHEWHART procedure, 1923
VZERO2 option
SHEWHART procedure, 1923

W

WAXIS= option
CUSUM procedure, 1923
MACONTROL procedure, 1923
PARETO procedure, 995, 1028
SHEWHART procedure, 1923
WBARLINE= option
PARETO procedure, 995, 1028
WEBOUT= option
CUSUM procedure, 1923
MACONTROL procedure, 1923
SHEWHART procedure, 1923
WEIGHT= option
MACONTROL procedure, 789
PARETO procedure, 995, 1028
WGRID2= option
PARETO procedure, 995, 1029
WGRID= option
CUSUM procedure, 1923
MACONTROL procedure, 1923
PARETO procedure, 995, 1028
SHEWHART procedure, 1923
WHERE statement
SHEWHART procedure, 1970–1973
WLIMITS= option
CUSUM procedure, 550
MACONTROL procedure, 1923
SHEWHART procedure, 1923
WMASK= option
CUSUM procedure, 550
WNEEDLES= option
MACONTROL procedure, 1924
SHEWHART procedure, 1924
WOVERLAY2= option
SHEWHART procedure, 1924
WOVERLAY= option
SHEWHART procedure, 1924
WSTARCIRCLES= option
CUSUM procedure, 1924
MACONTROL procedure, 1924
SHEWHART procedure, 1924, 1951
WSTARS= option
CUSUM procedure, 1924
MACONTROL procedure, 1924
SHEWHART procedure, 1924
WTESTS= option
SHEWHART procedure, 1924
WTREND= option
CUSUM procedure, 1924
MACONTROL procedure, 1924
SHEWHART procedure, 1924

X

XCHART statement, ANOM procedure,
See also ANOM procedure, XCHART statement
examples, advanced, 41
examples, introductory, 15
options summarized by function, 24–28, 30
overview, 15
syntax, 23
XCHART statement, CUSUM procedure,
See also CUSUM procedure, XCHART statement
examples, advanced, 569
examples, introductory, 521
notation, 551
overview, 521
syntax, 535
XCHART statement, SHEWHART procedure,
See also SHEWHART procedure, XCHART statement
examples, advanced, 1726
examples, introductory, 1692
options summarized by function, 1704, 1705, 1707–1711, 1714
overview, 1691
syntax, 1703
XRCHART statement,
See SHEWHART procedure, XRCHART statement
examples, advanced, 1777
examples, introductory, 1738
options summarized by function, 1751–1756, 1758, 1759
overview, 1737
syntax, 1750
XSCHART statement,
See SHEWHART procedure, XSCHART statement
examples, advanced, 1826
examples, introductory, 1790
options summarized by function, 1802–1805, 1807–1809, 1811
overview, 1789
syntax, 1801

XSYMBOL= option
MACONTROL procedure, 805, 860, 1924
SHEWHART procedure, 1780, 1924

Y

YPCT1= option
CUSUM procedure, 1925
MACONTROL procedure, 1925
SHEWHART procedure, 1925
YSCALE= option
SHEWHART procedure, 1925

Z

ZEROSTD option
SHEWHART procedure, 1925
ZONE2LABELS option
SHEWHART procedure, 1925
ZONE2VALUES option
SHEWHART procedure, 1926
ZONECHAR= option
SHEWHART procedure, 1926, 1990
ZONELABELS option
SHEWHART procedure, 1926, 1990
ZONES option
SHEWHART procedure, 1926, 1990
ZONES2 option
SHEWHART procedure, 1926
ZONEVALPOS= option
CUSUM procedure, 1926
SHEWHART procedure, 1926
ZONEVALUES option
SHEWHART procedure, 1926

Your Turn

If you have comments or suggestions about *SAS/QC® 9.1 User's Guide,* please send them to us on a photocopy of this page or send us electronic mail.

For comments about this book, please return the photocopy to

SAS Publishing
SAS Campus Drive
Cary, NC 27513
E-mail: **yourturn@sas.com**

For suggestions about the software, please return the photocopy to

SAS Institute Inc.
Technical Support Division
SAS Campus Drive
Cary, NC 27513
E-mail: **suggest@sas.com**