Statistics Toolbox Release Notes

Chapter 1, “Statistics Toolbox 5.1 Release Notes” describes the changes in this product since Version 5.0.2.

If you are upgrading from a version earlier than Version 5.0.2, you should also see

- Chapter 2, “Statistics Toolbox 5.0.2 Release Notes”
- Chapter 3, “Statistics Toolbox 5.0.1 Release Notes”
- Chapter 4, “Statistics Toolbox 5.0 Release Notes”
- Chapter 5, “Statistics Toolbox 4.1 Release Notes”
- Chapter 6, “Statistics Toolbox 4.0 Release Notes”
- Chapter 7, “Statistics Toolbox 3.0 Release Notes”

Printing the Release Notes. If you would like to print the Release Notes, you can link to a PDF version.

Contents

1 Statistics Toolbox 5.1 Release Notes

2 Statistics Toolbox 5.0.2 Release Notes

3 Statistics Toolbox 5.0.1 Release Notes

4 Statistics Toolbox 5.0 Release Notes

5 Statistics Toolbox 4.1 Release Notes

6 Statistics Toolbox 4.0 Release Notes

7 Statistics Toolbox 3.0 Release Notes

1

Statistics Toolbox 5.1 Release Notes

New Features

This section summarizes the new features and enhancements introduced in the Statistics Toolbox 5.1.

If you are upgrading from a release earlier than Release 14 with Service Pack 3, then you should also see “New Features” on page 2-2 in the Version 5.0.2 Release Notes and “New Features” on page 3-2 in the Version 5.0.1 Release Notes.

Partial Correlation

The new `partialcorr` function computes the correlation of one set of variables while controlling for a second set of variables.

Two New Univariate Probability Distributions

The Statistics Toolbox Version 5.1 provides support for two new distributions: the generalized extreme value distribution and the generalized Pareto distribution.

Generalized Extreme Value (GEV) Distribution

The GEV distribution combines the Gumbel, Frechet, and Weibull distributions into a single distribution , and can be used to model extremes in data.

The Statistics Toolbox 5.1 contains the following new functions for analyzing GEV distributions:

- `gevcdf` – Compute the cdf of a GEV distribution.
- `gevfit` – Compute parameter estimates and confidence intervals for GEV data.
- `gevinv` – Compute the inverse cdf of a GEV distribution.
- `gevlike` – Compute the negative log-likelihood for the GEV distribution.
- `gevpdf` – Compute the pdf of a GEV distribution.
- `gevrnd` – Generate random numbers from a GEV distribution.

- `gevstat` – Compute the mean and variance of a GEV distribution.

Generalized Pareto (GP) Distribution

The GP distribution can be used to model the tails of data.

The Statistics Toolbox 5.1 contains the following new functions for analyzing GP distributions:

- `gpcdf` – Compute the cdf of a GP distribution.
- `gpfit` – Compute parameter estimates and confidence intervals for GP data.
- `gpinv` – Compute the inverse cdf of a GP distribution.
- `gplike` – Compute the negative log-likelihood for the GP distribution.
- `gppdf` – Compute the pdf of a GP distribution.
- `gprnd` – Generate random numbers from a GP distribution.
- `gpstat` – Compute the mean and variance of a GP distribution.

New Hypothesis Tests

Chi-Square Goodness-of-Fit Test

The new `chi2gof` function tests for the goodness of fit of observed data to a specified distribution.

Variance Tests

Three functions have been added to test variances in one sample, two samples, or multiple samples. The new functions are `vartest`, `vartest2`, and `vartestn`, respectively.

Ansari-Bradley Test

The new `ansaribradley` function tests the hypothesis that two independent samples come from the same distribution, against the hypothesis that they come from distributions with the same median and shape but different variances.

Tests of Randomness

The new `runstest` function tests the hypothesis that the input values are in a random order.

New and Enhanced Functionality for Inverse Prediction and Prediction Intervals

- The new `invpred` function estimates the inverse prediction for simple linear regression.
- The `polyconf` function can compute either simultaneous or nonsimultaneous intervals for a new observation or for the polynomial itself. You can now enter optional arguments as parameter name/value pairs.
- The `grpstats` function can now compute a wider variety of descriptive statistics for grouped data. Choices include the mean, standard error of the mean, number of elements, group name, standard deviation, variance, confidence interval for the mean, and confidence interval for the new observation. You can also apply any other descriptive statistics function, or one you write yourself, to the grouped data.

Survival Analysis

The new `coxphfit` function fits the input data to Cox's proportional hazards regression, a distribution-free method for predicting survival as a function of other variables.

Enhanced Plotting Usability of ksdensity and ecdf

Both the `ksdensity` and the `ecdf` functions will now plot the results when no output arguments are specified.

New and Updated Demos

The Statistics Toolbox contains the following new demos for Version 5.1:

- Fitting a Univariate Distribution Using Cumulative Probabilities
- Curve Fitting and Distribution Fitting

- Pitfalls in Fitting Nonlinear Models by Transforming to Linearity
- Fitting an Orthogonal Regression Using Principal Components Analysis
- Weighted Nonlinear Regression
- Modelling Tail Data with the Generalized Pareto Distribution

The following demo has been updated for Version 5.1:

- Modelling Data with the Generalized Extreme Value Distribution

2

Statistics Toolbox 5.0.2 Release Notes

New Features

This section summarizes the new features and enhancements introduced in the Statistics Toolbox 5.0.2.

If you are upgrading from a release earlier than Release 14 with Service Pack 2, then you should also see “New Features” on page 3-2 in the Version 5.0.1 Release Notes and “New Features” on page 4-2 in the Version 5.0 Release Notes.

Cophenetic Correlation

The `cophenet` function computes the cophenetic correlation coefficient for a hierarchical cluster tree. This is the correlation between the cophenetic distances obtained from the tree and the original distances (or dissimilarities) used to construct the tree. Thus it is a measure of how faithfully the tree represents the dissimilarities among observations. Now in Version 5.0.2, the function also returns a second output that is the vector of cophenetic distances.

Major Bug Fixes

The Statistics Toolbox Version 5.0.2 includes important bug fixes made since Version 5.0.1. You can see a list of major Version 5.0.2 bug fixes on the MathWorks Web site.

In addition to the major bug fixes link listed above, you can view major bug fixes made in R14 with Service Pack 2 for Statistics Toolbox 5.0.1 using the Bug Reports interface on the MathWorks Web site.

Note Note: If you are not already logged in to the Access Login, when you link to the Bug Reports interface, you will be prompted to log in or create an Access Login account.

After you are logged in, use the Bug Reports link. You will see the bug report list for the Statistics Toolbox. The report is sorted with fixed bugs listed first, then open bugs.

If you are viewing these release notes in PDF form on the MathWorks Web site, please refer to the HTML form of the release notes on the MathWorks Web site and use the link provided.

If you are upgrading from a version earlier than Version 5.0.1, you should also see the Version 5.0.1 “Major Bug Fixes” on page 3-3.

3

Statistics Toolbox 5.0.1 Release Notes

New Features

This section summarizes the new features and enhancements introduced in the Statistics Toolbox 5.0.1.

This section covers the following topics:

New nancov Function

The new function `nancov` estimates the covariance matrix of a data set having missing values encoded as `NaN`. An option allows you to omit any row containing `NaN`, or to use the nonmissing information in that row when possible. Other functions for ignoring missing data include `nanmean` and `nanvar`.

regstats Function Returns Two New Statistics

The function `regstats` now returns the following two new statistics:

- `'rsquare'` – R-square statistic
- `'adjrsquare'` – Adjusted R-square statistic

Major Bug Fixes

The Statistics Toolbox 5.0.1 includes several bug fixes made since Version 5.0. This section describes the particularly important Version 5.0.1 bug fixes.

If you are viewing these Release Notes in PDF form, please refer to the HTML form of the Release Notes, using either the Help browser or the MathWorks Web site and use the link provided.

4

Statistics Toolbox 5.0 Release Notes

New Features

This section summarizes the new features and enhancements introduced in the Statistics Toolbox 5.0.

If you are upgrading from a release earlier than Release 13 with Service Pack 1, then you should also see “New Features” on page 5-2 in the Statistics Toolbox 4.1 Release Notes.

This section covers the following topics:

Distribution Fitting Tool

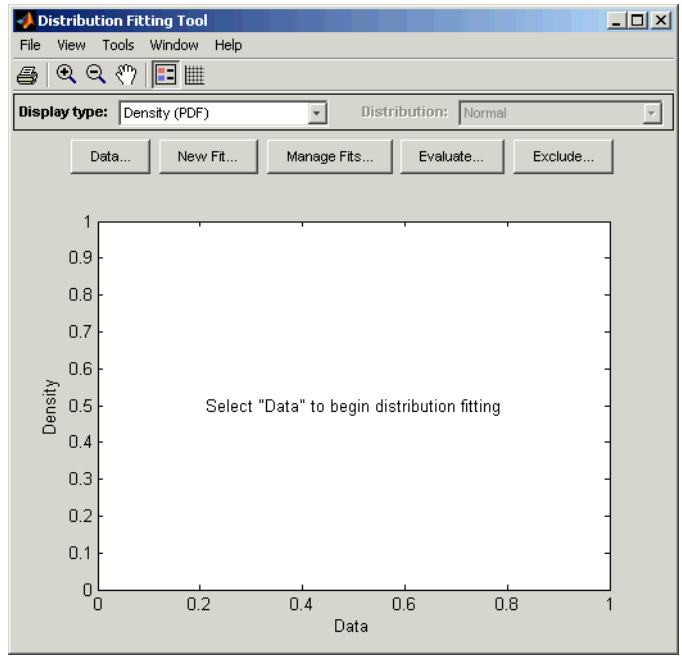
The Statistics Toolbox 5.0 provides a new graphical tool, the Distribution Fitting Tool, for fitting distributions to data. With the Distribution Fitting Tool, you can

- Fit distributions to data you import from the MATLAB workspace
- Plot distributions with the data
- Manage several different fits
- Evaluate distributions at various points

To start the tool, enter

```
dfittool
```

This opens the Distribution Fitting Tool, as shown in the following figure.



For more information, see Distribution Fitting Tool in the online Statistics Toolbox documentation.

New Functions

The Statistics Toolbox 5.0 contains the following new function:

- `andrewsplot` – Create an Andrews plot for multivariate data.
- `biplot` – Create a biplot of variable-factor coefficients from a factor analysis or a principal components analysis.
- `corr` – Compute linear or rank correlation, and p-values.
- `glyphplot` – Plot stars or Chernoff faces for multivariate data.
- `hist3` – Create a 3-dimensional histogram of bivariate data.
- `mdscale` – Perform nonmetric and metric multidimensional scaling.

- `mlecov` – Approximate the asymptotic covariance matrix for the maximum likelihood parameter estimates of a specified distribution.
- `nanvar` – Compute variance ignoring `NaN`s.
- `parallelcoords` – Create a parallel coordinates plot for multivariate data.
- `probplot` – Create a probability plot.
- `quantile` – Compute quantiles of a sample.
- `rotatefactors` – Rotate a factor analysis or principal components analysis solution.

N-Dimensional Support for Functions

The following functions now accept N-dimensional arrays as inputs.

Descriptive Statistics

- `geomean`
- `harmmean`
- `iqr`
- `kurtosis`
- `mad`
- `moment`
- `nanmax`
- `nanmean`
- `nanmedian`
- `nanmin`
- `nanstd`
- `prctile`
- `range`
- `trimmean`

Random Number Generators

The functions listed below can now return N-dimensional arrays as outputs. For example,

```
R = poissrnd(lambda,3,5,4)
```

or

```
R = poissrnd(lambda, [3,5,4])
```

returns a 3-by-5-by-4 array of numbers chosen from the Poisson distribution with parameter `lambda`.

These functions can now also accept an N-dimensional array as an input. For example, if `lamda` is an N-dimensional array of parameters,

```
R = poissrnd(lamda)
```

returns an N-dimensional array of the same size as `lambda`.

- `betarnd`
- `binornd`
- `chi2rnd`
- `evrnd`
- `exprnd`
- `frnd`
- `gamrnd`
- `geornd`
- `hygernd`
- `lognrnd`
- `mvnrnd`
- `mvtrnd`
- `nbinrnd`
- `ncfrnd`

- nctrnd
- ncx2rnd
- normrnd
- poissrnd
- random
- raylrnd
- trnd
- unidrnd
- unifrnd
- wblrnd

Enhancements to Function mle

The function mle has the following enhancements for Version 5.0:

- mle now uses the following new syntax:
  - phat = mle(data) returns maximum likelihood estimates (MLEs) for the parameters of a normal distribution, computed using the sample data in the vector data.
  - phat = mle(data,'distribution', dist) returns MLEs for the parameters of the distribution dist.
  - phat = mle(data, 'name1', value1, 'name2', value2,...) takes additional options specified as name/value pairs. Type help mle to see the available options.
- mle now accepts the input arguments 'logn' and 'lognormal' as values for the 'distribution' option. For example,

  ```
  phat = mle(data,'distribution', 'logn')
  ```

  returns parameter estimates for a lognormal distribution of data.
- mle now accepts custom distributions. For example,

  ```
  phat = mle(data, 'pdf', @mypdf)
  ```

returns MLEs for the distribution given by the probability density function `mypdf`, which you can write as an M-file. Type `help mle` to learn about other ways to define custom distributions.

Enhancement to Other Functions

The Statistics Toolbox 5.0 includes enhancements to following functions:

- `mad` now accepts a flag to indicate whether it should compute the mean of the absolute deviations from the mean of the data, or the median of the absolute deviations from the median of the data. The command

  ```
  Y = mad(X,1)
  ```

  computes `Y` based on medians. The command

  ```
  Y = mad(X,0)
  ```

  which is the same as `mad(X)`, computes `Y` based on means.

- `pdist` now accepts the input argument `'chebychev'`. The command

  ```
  Y = pdist(X, 'chebychev')
  ```

  computes a vector `Y` containing the pairwise distances between the rows of `X`, using Chebychev distance, the maximum coordinate difference between two rows.

- `ncx2rnd` now accepts noninteger parameters. In previous releases, if you called `ncx2rnd` with noninteger parameters, MATLAB returned a warning. In Version 5.0, `ncx2rnd` accepts noninteger parameters for the noncentral Chi-squared distribution.

- `princomp` now has an additional input argument `'econ'`. When you call `princomp` with the syntax

  ```
  princomp(X, 'econ')
  ```

  the function only returns outputs corresponding to the components with nonzero variance. This only affects the outputs when the matrix `X` has at least as many columns as rows. You can use the `'econ'` flag to make computations feasible when the data matrix `X` has a large number of variables (columns).

Major Bug Fixes

There are no major bug fixes in the Statistics Toolbox 5.0 since Version 4.1.

If you are upgrading from a release earlier than Release 13 with Service Pack 1, then you should see the bug fixes summary for the Statistics Toolbox 4.1 Release Notes. If you are viewing these Release Notes in PDF form, please refer to the HTML form of the Release Notes, using either the Help browser or the MathWorks Web site and use the link provided.

5

Statistics Toolbox 4.1 Release Notes

New Features

This section summarizes the new features and enhancements introduced in the Statistics Toolbox 4.1.

If you are upgrading from a release earlier than Release 13, then you should also see "New Features" on page 6-2 in the Statistics Toolbox 4.0 Release Notes.

The Statistics Toolbox 4.1 includes the following new features and enhancements:

Improved N-way Analysis of Variance

The Statistics Toolbox 4.1 includes enhancements to the function `anovan`, which performs N-way analysis of variance. You can now specify that grouping variables in an ANOVA model are random effects, using the syntax

```
p = anovan(x, group, 'random', rand_effects)
```

where `rand_effects` is a vector specifying which grouping variables are random effects. See "ANOVA with Random Effects" for an example of this feature.

`anovan` also uses a new property name/property value syntax for several of its input arguments. For example, if you want the function to use an interaction model, you can now enter

```
p = anovan(x, group, 'model', 'interaction')
```

The Version 4.0 syntax for this command,

```
p = anovan(x, group, 2)
```

still works, but we recommend that you now use the new syntax.

See the reference page for `anovan` for details.

Hidden Markov Model Functions

The Statistics Toolbox 4.1 contains five new functions for analyzing hidden Markov models. Hidden Markov models are flexible probabilistic models that have applications in areas such as speech recognition, machine vision and bioinformatics. The new functions are

- `hmmdecode` – Calculates the posterior state probabilities of a sequence.
- `hmmgenerate` – Generates a sequence for a hidden Markov model.
- `hmmestimate` – Estimates the parameters for a hidden Markov model.
- `hmmtrain` – Calculates the maximum likelihood estimate of hidden Markov model parameters.
- `hmmviterbi` – Calculates the most probable state path for a hidden Markov model sequence.

See "Hidden Markov Models" for descriptions of these functions.

New Functions for Weibull Distributions

The Statistics Toolbox 4.1 contains the following new functions for analyzing Weibull distributions:

- `wblcdf` – Compute the cdf of a Weibull distribution.
- `wblfit` – Compute parameter estimates and confidence intervals for Weibull data.
- `wblinv` – Compute the inverse of the Weibull distribution.
- `wbllike` – Compute the negative log-likelihood for Weibull data.
- `wblpdf` – Compute the pdf of a Weibull distribution.
- `wblplot` – Create Weibull probability plot.

- `wblrnd` – Generate random numbers from a Weibull distribution.
- `wblstat` – Compute the mean and variance of a Weibull distribution.

These functions replace the Version 4.0 Weibull distribution functions, whose names began with `weib` instead of `wbl`. While code containing the Version 4.0 Weibull functions will continue to function correctly, we recommend that you use the new functions in Version 4.1.

New Functions for Extreme Value Distributions

The Statistics Toolbox 4.1 contains the following new functions for analyzing extreme value distributions:

- `evcdf` – Compute the cdf of a extreme value distribution.
- `evfit` – Compute parameter estimates and confidence intervals for extreme value data.
- `evinv` – Compute the inverse of the extreme value distribution.
- `evlike` – Compute the negative log-likelihood for extreme value data.
- `evpdf` – Compute the pdf of a extreme value distribution.
- `evrnd` – Generate random numbers from a extreme value distribution.
- `evstat` – Compute the mean and variance of a extreme value distribution.

See "Extreme Value Distribution" for more information.

Distribution Functions that Now Accept Censored Data

The following distribution functions now accept an input argument for censored data, which enables you to analyze survival data:

- `evfit` and `evlike`
- `expfit` and `explike`
- `lognfit` and `lognlike`
- `normfit` and `normlike`

- `wblfit` and `wbllike`

New Stepwise Regression Tools and Improved GUIs

The Statistics Toolbox 4.1 includes the following additions and enhancements to its stepwise regression tools:

- The new function `stepwisefit` enables you to perform stepwise regression from the command line.
- The new function `addedvarplot` enables you to create added-variable plots for stepwise regression.
- Enhancements to the GUIs for the following functions make them easier to use:
  - `stepwise`
  - `disttool`
  - `randtool`
  - `regstats`
  - `robustdemo`

See “Stepwise Regression Demo” for an example.

New Utility Functions for Statistics Options

The Statistics Toolbox 4.1 contains two new utility functions for creating and editing the `options` structure that several of the distribution fitting functions use to compute maximum likelihood estimates.

- `statget` – Get parameter values from a statistics options structure
- `statset` – Create or edit a statistics options structure

Major Bug Fixes

The Statistics Toolbox 4.1 includes several bug fixes made since Version 4.0. This section describes the particularly important Version 4.1 bug fixes.

If you are viewing these Release Notes in PDF form, please refer to the HTML form of the Release Notes, using either the Help browser or the MathWorks Web site and use the link provided.

If you are upgrading from a release earlier than Release 13, then you should also see “Major Bug Fixes” on page 6-8 in the Statistics Toolbox 4.0 Release Notes.

Upgrading from an Earlier Release

This section describes the upgrade issues involved in moving to the Statistics Toolbox 4.1 from Version 4.0.

New Format of Terms Output for anovan Might Cause Errors in Existing Code

In the Statistics Toolbox 4.1, the function `anovan` returns the output argument `terms` as a matrix. In Version 4.0, `anovan` returned `terms` as a vector. If you have existing code that does the following three, it will no longer work in Version 4.1:

1. Calls `anovan` with the output `terms`.
2. Modifies `terms` by adding or removing entries.
3. Subsequently uses `terms` as the input for the `model` argument of `anovan`, using the Version 4.0 syntax.

For example, the following code no longer works in Version 4.1.

```
load carbig
[p, t, stats, terms] = anovan(MPG, {Cylinders Origin Model_Year});
terms = terms(1:2) % Remove last entry of vector terms
anovan(MPG, {Cylinders Origin Model_Year},terms);
```

To fix this code, change lines 3 and 4 as follows:

```
terms = terms(1:2, :) % Remove last row of matrix terms
anovan(MPG, {Cylinders Origin Model_Year},'model', terms);
```

in Existing Code 1-11

6

Statistics Toolbox 4.0 Release Notes

New Features

This section summarizes the new features and enhancements introduced in the Statistics Toolbox 4.0.

If you are upgrading from a release earlier than Release 12.0, then you should also see “New Features” on page 7-2 in the Statistics Toolbox 3.0 Release Notes.

Multivariate Analysis

Cluster Analysis

The new `kmeans` function performs K-means clustering and supports five different distance measures. The new function `silhouette` plots silhouettes of clusters created using either K-means or hierarchical clustering methods. The `pdist` function now allows several new distance measures and is more efficient for large datasets.

Factor Analysis

The new `factoran` function fits a Common Factor Analysis model using maximum likelihood, including rotation of the estimated factor loadings and estimation of factor scores.

Multidimensional Scaling and Procrustes Analysis

The new `cmdscale` function performs classical (metric) Multidimensional Scaling, to create a configuration of points in Euclidean space solely from distance data. The new function `procrustes` performs orthogonal Procrustes rotations to match one set of points onto another.

Canonical Correlation Analysis

The new function `canoncorr` performs Canonical Correlation Analysis, to find the subsets of variables in two datasets that best correlate with each other.

Discriminant Analysis

The `classify` function now supports three types of discrimination (linear, quadratic, and Mahalanobis) and allows specification of prior probabilities.

'`linear`' is now the default, and you must specify `'mahalanobis'` to duplicate the behavior of the previous version.

Nonlinear Regression Models

Classification and Regression Trees

A collection of new functions (`treefit`, `treeprune`, `treedisp`, `treetest`, `treeval`) performs classification and regression using decision trees. These functions fit trees to data, display them, prune them, compute error rates for them using test data or cross-validation, and apply them to new data.

Probability Distributions

Several new functions support the generation of random samples from multivariate distributions. There are functions for generating random matrices from the Wishart (`wishrnd`) or inverse Wishart (`iwishrnd`) distributions. Other functions (`lhsdesign`, `lhsnorm`) use latin hypercube sampling methods to generate samples from the multivariate uniform and normal distributions. In addition there have been improvements in other probability functions, particularly those for the negative binomial distribution. Finally, a new function (`mvnpdf`) computes the probability density function for the multivariate Normal distribution.

Descriptive Statistics

Density Estimation

The new `ksdensity` function produces a nonparametric density estimate using a kernel smoothing technique.

Empirical Cumulative Distribution

The new `ecdf` function computes the empirical cumulative distribution function (cdf) and confidence bounds for it. For censored data (common in survival analysis), it computes the Kaplan-Meier estimate of the cdf.

Design of Experiments

Response Surface Designs

New functions support two commonly used designs: central composite designs (`ccdesign`) and Box-Behnken designs (`bbdesign`). Central composite designs fit a full quadratic model and can have three or five levels of each factor. `ccdesign` supports the three types, circumscribed, inscribed and faced.

Box-Behnken designs are rotatable designs that also fit a full quadratic model but use just three levels of each factor.

D-Optimal Designs

The D-optimal design generation functions are faster than in the past. In addition, the two new functions `candgen` and `candexch` provide more control over the row-exchange algorithm for design generation.

Function Summary

Version 4.0 of the Statistics Toolbox provides the following:

- "New Functions" on page 6-4
- "Statistics Functions with New or Changed Capabilities " on page 6-6

New Functions

Function	Purpose
`bbdesign`	Generate Box-Behnken design
`candexch`	D-optimal design from candidate set using row exchanges
`candgen`	Generate candidate set for D-optimal design

Function	Purpose
canoncorr	Canonical correlation analysis
ccdesign	Generate central composite design
cmdscale	Classical multidimensional scaling
ecdf	Empirical (Kaplan-Meier) cumulative distribution function
factoran	Perform Factor Analysis by maximum likelihood
iwishrnd	Generate inverse Wishart random matrix
kmeans	K-means clustering
ksdensity	Compute a probability density estimate using a kernel smoothing method
lhsdesign	Generate a latin hypercube sample
lhsnorm	Generate a multivariate normal random matrix using latin hypercube sampling
mvnpdf	Multivariate normal probability density function (pdf)
nbinfit	Parameter estimates and confidence intervals for negative binomial data
procrustes	Procrustes Analysis
silhouette	Silhouette plot for clustered data
treefit	Fit a tree-based model for classification or regression.
treeprune	Produce a sequence of subtrees by pruning.
treedisp	Show classification or regression tree graphically
treetest	Compute error rate for tree
treeval	Compute fitted value for decision tree applied to data
wishrnd	Generate Wishart random matrix

Statistics Functions with New or Changed Capabilities

Function	Enhancement or Change
`classify`	A new syntax lets you specify the type of discriminant function as `'linear'` (default), `'quadratic'`, or `'mahalanobis'`. Specify `'mahalanobis'` to duplicate the behavior of the previous version. Another new syntax enables you to specify prior probabilities for the groups. A new output returns an estimate of the misclassification error rate.
`cluster`	Now also allows clustering based on distance measures. A new syntax also enables you to specify values for these parameters:
'cutoff'	Cutoff for inconsistent and distance measure
'maxclust'	Maximum number of clusters to form
'criterion'	Either `'inconsistent'` or `'distance'`
'depth'	Depth for computing inconsistent values
The old syntax still works but is undocumented.	
`clusterdata`	`clusterdata(Z,'param1',val1,'param2',val2,...)` now enables you to specify parameters that `clusterdata` uses in calling `pdist`, `linkage`, and `cluster`:
'distance'	Any of the distance metric names allowed by `pdist`
'linkage'	Any of the linkage methods allowed by `linkage`
'cutoff'	Cutoff for inconsistent and distance measure
'maxclust'	Maximum number of clusters to form
'criterion'	Either `'inconsistent'` or `'distance'`
'depth'	Depth for computing inconsistent values

`cordexch`
`daugment`
`dcovary`
`rowexch`
	A new syntax provides more control over design generation through a set of parameter-value pairs.
	`function(...,'param1',value1,'param2',value2,...)`
	Valid parameters are:

Function	Enhancement or Change
'display'	Controls display of iteration counter.
'init'	Specifies an initial design. The default is a randomly selected set of points.
'maxiter'	Specifies the maximum number of iterations. The default is 10.
`corrcoef` (MATLAB)	Provides three new syntaxes: `[R,P] = corrcoef(...)` returns `P`, a matrix of p-values for testing the hypothesis of no correlation. `[R,P,RLO,RUP] = corrcoef(...)` returns matrices `RLO` and `RUP` which contain lower and upper bounds for a 95% confidence interval for each coefficient. `[...]=corrcoef(...,'param1',val1,'param2',val2,...)` accepts parameter-value pairs that enable you to override the default confidence interval, and specify how to treat rows of `X` that contain `NaN`s.
`nbincdf`, `nbininv`, `nbinpdf`, `nbinrnd`, `nbinstat`	Consistent with a more general interpretation of the negative binomial, these functions now accept any positive value, including nonintegers, for the size parameter `R`.
`pdist`	Provides four new metrics for calculating the pairwise distance between observations: `'cosine'`, `'correlation'`, `'hamming'`, and `'jaccard'`. It now also accepts a function handle to a user-defined distance function.
`regstats`	A new syntax `stats = regstats(responses,DATA,model,whichstats)` creates an output structure `stats` containing the statistics listed in `whichstats`. `whichstats` can be a single name or a cell array of names. The list of available statistics remains the same.

Major Bug Fixes

The Statistics Toolbox 4.0 includes several bug fixes made since Version 3.0. This section describes the particularly Important Version 4.0 bug fixes.

If you are viewing these Release Notes in PDF form, please refer to the HTML form of the Release Notes, using either the Help browser or the MathWorks Web site and use the link provided.

Upgrading from an Earlier Release

This section describes the upgrade issues involved in moving from the Statistics Toolbox 3.0 to Version 4.0.

Linear and Quadratic Discriminant Analysis Added to classify

The algorithm that was previously implemented in `classify` used the Mahalanobis distance between sample points and training groups, with stratified estimates of covariance. The new implementation adds the standard algorithms for linear (default) and quadratic discriminant analysis. Set `'type'` to `'mahalanobis'` in Version 4.0 (Release 13) to duplicate the behavior of the previous version.

Use playshow Command to Run glmdemo

Starting in Release 13, to run slideshow style demos such as `glmdemo` from the command line, you must use the `playshow` command. For example,

```
playshow glmdemo
```

You can continue to run other styles of demos from the command line by typing just the demo name. `glmdemo` is the only slideshow style demo in the Statistics Toolbox Version 4.0.

7

Statistics Toolbox 3.0 Release Notes

New Features

This section introduces the new features and enhancements added in the Statistics Toolbox 3.0 since the Statistics Toolbox 2.2 (Release 11.0).

Summary of Enhancements

Expanded Support for Linear Models

Version 3.0 expands the Statistics Toolbox support for linear models in general, and analysis of variance in particular. The following are the major changes for the Statistics Toolbox 3.0:

- Improvements in one-way analysis of variance (`anova1`)
- Higher-way analysis of variance (`anovan`)
- Analysis of covariance (`aoctool`)
- Multiple comparisons of means or other estimates (`multcompare`)
- Multivariate analysis of variance (`manova1`, `manovacluster`)
- Graphics functions useful for examining data used for multivariate analysis of variance (`gscatter`, `gplotmatrix`, `gname`)
- Response surface fitting with multiple responses (`rstool`)
- Nonparametric analysis of variance (`friedman`, `kruskalwallis`)
- More flexible calculation of confidence bounds (`polytool`, `nlintool`, `nlpredci`)

Other Enhancements

In addition, the following changes do not involve linear models:

- Generalized linear models (`glmfit`, `glmval`)
- Robust regression (`robustfit`, `polytool`)
- Distribution testing and plotting (`cdfplot`, `lillietest`, `kstest`, `kstest2`)
- Fractional factorial design generation (`fracfact`)

- Importing numeric and text data from tab-delimited files (`tdfread`)
- More flexible handling of grouping variables (`boxplot`, `grpstats`)
- Multivariate t random number generation (`mvtrnd`) and improvements to other t distribution functions

Numerous other functions received enhancements, as described in the following sections:

- "New Functions" on page 7-3
- "New Demos" on page 7-4
- "New Sample Data Files" on page 7-5
- "Updated Functions for ANOVA-Type Tables" on page 7-5
- "Other Updated Functions" on page 7-6

New Functions

The following functions have been added to the Statistics Toolbox 3.0.

Function	Description
`anovan`	N-way Analysis of Variance (ANOVA)
`aoctool`	Interactive plot for fitting and predicting analysis of covariance models
`cdfplot`	Plot of empirical cumulative distribution function
`fracfact`	Generate fractional factorial design from generators
`friedman`	Friedman's nonparametric two-way Analysis of Variance (ANOVA)
`glmfit`	Generalized linear model fitting
`glmval`	Compute predictions for generalized linear model
`gplotmatrix`	Plot matrix of scatter plots by group
`gscatter`	Scatter plot by group
`jbtest`	Jarque-Bera test for goodness-of-fit to a normal distribution

Function	Description
kruskalwallis	Kruskal-Wallis nonparametric one-way Analysis of Variance (ANOVA)
kstest	Kolmogorov-Smirnov test of the distribution of one sample
kstest2	Kolmogorov-Smirnov test to compare the distribution of two samples
lillietest	Lilliefors test for goodness-of-fit to a normal distribution
manova1	One-way Multivariate Analysis of Variance (MANOVA)
manovacluster	Plot dendrogram showing group mean clusters after MANOVA
multcompare	Multiple comparison test of means or other estimates
mvtrnd	Random matrices from the multivariate t distribution
robustfit	Robust regression
tdfread	Read file containing tab-delimited numeric and text values

New Demos

glmdemo

The glmdemo function is a slideshow-style demo of generalized linear model fitting.

Note To run glmdemo from the command line in Version 4.0, Release 13, you must type playshow glmdemo. In Version 3.0, Release 12, you need only type glmdemo.)

---

robustdemo

The `robustdemo` function demonstrates robust fitting. The function graphs (*x*,*y*) data with an outlier, and shows how the least squares and robust fits differ. You can move points with the mouse, and see how the two fits change. You can also display the least squares leverage and the robust weight for each point. You can also provide input data instead of using the built-in example.

New Sample Data Files

carbig

The `carbig` data file is a large dataset on cars from the 70s and 80s.

carsmall

The `carsmall` data file is a subset of `carbig`, containing cars from just three model years.

Updated Functions for ANOVA-Type Tables

Several functions display tables, such as ANOVA tables, in a figure window. These figure windows now have a new **Copy Text** option on the **Edit** menu. You can use this option to copy the table as tab-delimited text into Microsoft Excel, Microsoft Word, or other applications. These two functions that produce such tables have been updated:

- `anova1`
- `anova2`

The changes to each of these functions are described below.

Note The `aoctool`, `friedman`, and `kruskalwallis` functions are new functions added in the Statistics Toolbox 3.0; these functions also display ANOVA-type tables.

---

anova1

```
[p,table,stats] = anova1(x,group,'displayopt')
```

- New output `table` is a cell array of the ANOVA table values, including row and column labels.
- Now returns a `stats` output structure useful for performing multiple comparisons (see `multcompare` for more information).
- New input `'displayopt'` is `'off'` to omit the table and boxplot display, or `'on'` (the default) to display the table and boxplot.
- If `x` is a matrix, `group` can now be a character array or cell array of strings with one row for each column of `x`. The boxes in the boxplot are then labeled using the rows of `group`.
- If `x` is a vector, `group` can be a vector of integers or a character array or cell array of strings with one row for each element of `x`. The boxes in the boxplot are labeled with values from `group`.
- P-value added to both `table` and to the table display.
- Now accepts group numbers that are not of the form 1, ..., *g*.

anova2

```
[p,table,stats] = anova2(x,reps,'displayopt')
```

- Now returns `table` as a second output.
- The additional input `'displayopt'` can be used to suppress the ANOVA table display.
- The additional output `stats` can be used as input to `multcompare` to perform multiple comparisons of row or column means.

Other Updated Functions

Linear Model Functions

Linear model functions (e.g., `anova1`, `polyval`, etc.) ignore observations with `NaN` value in the `X` or `Y` input.

betafit

```
[phat,pci] = betafit(x,alpha)
```

The `betafit` function now:

- Removes `NaN` data before fitting
- Issues an error message if there are any `0` or `1` values
- Issues an error message if `x` is constant

boxplot

```
boxplot(x,notch,sym,vert,whis)
boxplot(x,g,notch,sym,vert,whis)
```

The second syntax for `boxplot` above is new. The first syntax displays a box for each column of the `x` matrix. The second syntax displays a box for each level of the grouping variable `g`. In addition, `g` can be a cell array of grouping variables to produce a separate box for each unique combination of grouping variable levels. See `grpstats`.

cluster

```
T = cluster(Z,cutoff,depth,flag)
```

The `cluster` function adds a flag argument which overrides the default meaning of the `cutoff` argument. If `flag` is `'inconsistent'`, then `cutoff` is interpreted as a threshold for the inconsistency coefficient. If `flag` is `'clusters'`, then `cutoff` is the maximum number of clusters.

crosstab

```
[table,chi2,p,labels] = crosstab(col1,col2,...)
```

The `crosstab` function now accepts any number of inputs, not just two. Each input can be a numeric vector, a string array, or a cell array of strings. (In the previous release each input had to be a vector of positive integers taking

values 1, ..., *g* for some *g*.) If there are *v* input variables, the output `table` is a *v*-dimensional array, with `table(i,j,k,...)` counting the number of times that the first argument takes its *i*th value, that the second argument takes its *j*th value, that the third argument takes its *k*th value, and so on.

For the case of two positive integer input arguments, the function yields the same results as the previous release unless there are missing integers (i.e., not all of 1, ..., *g* appear in the input). In that case, the previous release would have produced a divide-by-zero warning and would have generated a row or column of zeros in `table`. The new version simply does not consider that category, so it does not reserve zeros for it.

As in the previous release, `chi2` is a chi-square statistic for testing independence, and `p` is its p-value. In this release, `table` can be other than a two-dimensional table, and the test is that all dimensions are independent.

The `labels` output is a cell array with one column for each input argument. The column lists the values of that input. Revisiting the example above, `table(i,j,k,...)` counts the number of times that the first argument takes the value `labels{i,1}`, that the second argument takes the value `labels{j,2}`, that the third argument takes the value `labels{k,3}`, and so on.

ewmaplot

```
h = ewmaplot(data,lambda,alpha,specs)
```

The `ewmaplot` default for `alpha` changed to 0.27% to conform to the standard ewma chart definition.

grpstats

```
[means,sem,counts,gname] = grpstats(x,group)
```

The `grpstats` argument `group` is no longer restricted to be a vector of integers. It can be a grouping variable that is a numeric vector, a string matrix, or a cell array of strings. In addition it can be a cell array containing multiple group vectors. The function computes statistics on groups defined by unique combinations of levels of the grouping variables. The new output

gname is a cell array with one row per group and one column per grouping variable. Elements of means, sem, and counts are statistics calculated for the group defined by values in the corresponding row of gname. Examples include

```
[m,s,c] = grpstats(x,g1);
[m,s,c,gnames] = grpstats(x,{g1 g2});
```

nlinfit

```
[beta,r,J] = nlinfit(X,y,fun,beta0)
```

The nlinfit function now accepts inline functions and function handles (@FF) in addition to the text strings ('FF') accepted in the past for input fun.

nlintool

```
nlintool(x,y,fun,beta0,alpha,'xname','yname')
```

The interface invoked with the nlintool function now:

- Adds a new menu option to compute different types of confidence intervals. Intervals can be simultaneous (provide a specified confidence level over all x values simultaneously) or nonsimultaneous (provide that level for a single predetermined x value). They can apply to the estimated regression function only (not taking account any variability from a new observation) or to a prediction for a new observation (taking its variability into account).
- Accepts inline functions and function handles (@FF) in addition to the text strings ('FF') accepted in the past for input fun.

nlpredci

```
ypred = nlpredci(fun,inputs,beta,r,J,alpha,'simopt','predopt')
```

The nlpredci function has new arguments that allow the same types of confidence intervals produced by nlintool.

norminv

```
x = norminv(p,mu,sigma)
```

The `norminv` function now returns `NaN` for each element of `p` that is `NaN`.

normplot

```
h = normplot(x)
```

The `normplot` function now strips `NaN` values individually from each column of `x`.

normrnd

```
r = normrnd(mu,sigma,m,n)
```

The `normrnd` function now returns the mean if `sigma` is `0`.

polytool

```
h = polytool(x,y,n,alpha,xname,yname)
```

The interface invoked by the `polytool` function has the following enhancements:

- Removes `x(j)` and `y(j)` if either is `NaN`, and display a warning when doing so.
- The **Method** menu provides the option of using robust (bisquare) fitting in place of least squares.
- The new **Bounds** menu option computes different types of confidence intervals. Intervals can be simultaneous (provide a specified confidence level over all `x` values simultaneously) or nonsimultaneous (provide that level for a single predetermined `x` value). They can apply to the estimated regression function only (not taking account any variability from a new observation) or to a prediction for a new observation (taking its variability into account).

prctile

```
y = prctile(x,p)
```

The `prctile` function now strips `NaN` values individually from each column of `x`.

qqplot

```
h = qqplot(x,y,pvec)
```

The `qqplot` function now:

- Strips `NaN` values individually from each column of `x` and `y`.
- If `y` is omitted, uses standard normal quantiles.

ranksum

```
[p,h,stats] = ranksum(x,y,alpha)
```

New `ranksum` output `stats` is a structure that always contains a field named `ranksum` whose value is the value of the rank sum statistic, and that for large samples contains a field named `zval` that is the value of the normal (Z) statistic used to compute the p-value `p`.

schart

```
[outliers,h] = schart(data,conf,specs)
```

The `schart` default for `conf` changed to 99.73% to conform to the standard s-chart definition.

signrank

```
[p,h] = signrank(x,y,alpha)
```

The `signrank` function has the following enhancements:

- If y is a scalar, extend it to the same length as x. This facilitates comparison of the median of one sample to a constant value.
- If x and y are the same, return p=1 and h=0.
- If p=alpha, now h=1 rather than h=0 (rejects hypothesis).
- New return value stats is a structure that always contains a field named signed rank whose value is the value of the signed rank statistic, and that for large samples contains a field named zval that is the value of the normal (Z) statistic used to compute the p-value p.

signtest

```
[p,h,stats] = signtest(x,y,alpha)
```

The signtest function has the following enhancements:

- If p=alpha, now h=1 rather than h=0 (rejects hypothesis)
- New return value stats is a structure that always contains a field named sign whose value is the value of the sign statistic, and that for large samples contains a field named zval that is the value of the normal (Z) statistic used to compute the p-value p. Z is signed, not an absolute value.

tcdf, tinv, tpdf, trnd, tstat

The tcdf, tinv, tpdf, trnd, and tstat functions now accept noninteger degrees of freedom.

ttest

```
[h,sig,ci,stats] = ttest(x,m,alpha,tail)
```

The ttest function has these enhancements:

- Added new output stats, the value of the *t* statistic and its degrees of freedom.
- Removes NaN from x before starting test.

ttest2

```
[h,sig,ci,stats] = ttest2(x,y,alpha,tail)
```

The `ttest2` function has these enhancements:

- Adds new output `stats`, the value of the *t* statistic and its degrees of freedom.
- If `tail` is `1` or `-1`, now `ci` has one endpoint set to `Inf` or `-Inf`.
- Removes `NaN` from `x` and `y` before starting test.

xbarplot

```
[outliers,h] = xbarplot(data,conf,specs,'sigmaest')
```

The `xbarplot` function has these enhancements:

- Changed default `conf` to 99.73%, to conform to the standard x-bar chart definition.
- Corrected calculation of control limits. With the new default `conf` the control limits are three-sigma limits.
- New input `sigmaest` specifies how to estimate sigma in the control limit calculation. The default is `'std'`, meaning estimate using the average of the subgroup standard deviations. The value `'variance'` uses a pooled variance estimate; this was the value used in previous releases. The value `'range'` uses the average of the subgroup ranges, and requires subgroups with no more than 25 observations.

ztest

```
[h,sig,ci,zval] = ztest(x,m,sigma,alpha,tail)
```

The `ztest` function has these enhancements:

- Adds new output `zval`, the value of the test statistic.
- If `tail` is `1` or `-1`, now `ci` has one endpoint set to `Inf` or `-Inf`.

- Removes `NaN` from `x` before starting test.