NAG Fortran Library Routine Document

G02BTF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

G02BTF updates the sample means and sums of squares and cross-products, or sums of squares and cross-products of deviations about the mean, for a new observation. The data may be weighted.

2 Specification

```
SUBROUTINE G02BTF(MEAN, M, WT, X, INCX, SW, XBAR, C, IFAIL)
INTEGER           M, INCX, IFAIL
real              WT, X(M*INCX), SW, XBAR(M), C((M*M+M)/2)
CHARACTER*1       MEAN
```

3 Description

G02BTF is an adaptation of West's WV2 algorithm; see West (1979). This routine updates the weighted means of variables and weighted sums of squares and cross-products or weighted sums of squares and cross-products of deviations about the mean for observations on m variables X_j, for $j = 1, 2, \ldots, m$. For the first $i-1$ observations let the mean of the jth variable be $\bar{x}_j(i-1)$, the cross-product about the mean for the jth and kth variables be $c_{jk}(i-1)$ and the sum of weights be W_{i-1}. These are updated by the ith observation, x_{ij}, for $j = 1, 2, \ldots, m$, with weight w_i as follows:

$$W_i = W_{i-1} + w_i, \quad \bar{x}_j(i) = \bar{x}_j(i-1) + \frac{w_i}{W_i}(x_j - \bar{x}_j(i-1)), \quad j = 1, 2, \ldots, m$$

and

$$c_{jk}(i) = c_{jk}(i-1) + \frac{w_i}{W_i}(x_j - \bar{x}_j(i-1))(x_k - \bar{x}_k(i-1))W_{i-1}, \quad j = 1, 2, \ldots, m;\ k = j, j+1, 2, \ldots, m.$$

The algorithm is initialised by taking $\bar{x}_j(1) = x_{1j}$, the first observation and $c_{ij}(1) = 0.0$.

For the unweighted case $w_i = 1$ and $W_i = i$ for all i.

4 References

Chan T F, Golub G H and Leveque R J (1982) *Updating Formulae and a Pairwise Algorithm for Computing Sample Variances* Compstat, Physica-Verlag

West D H D (1979) Updating mean and variance estimates: An improved method *Comm. ACM* **22** 532–555

5 Parameters

1: MEAN – CHARACTER*1 *Input*

On entry: indicates whether G02BTF is to calculate sums of squares and cross-products, or sums of squares and cross-products of deviations about the mean.

If MEAN = 'M', the sums of squares and cross-products of deviations about the mean are calculated.

If MEAN = 'Z', the sums of squares and cross-products are calculated.

Constraint: MEAN = 'M' or 'Z'.

2: M – INTEGER *Input*

On entry: the number, m, of variables.

Constraint: M ≥ 1.

3: WT – ***real*** *Input*

On entry: the weight to use for the current observation, w_i.

For unweighted means and cross-products set WT $= 1.0$. The use of a suitable negative value of WT, e.g., $-w_i$ will have the effect of deleting the observation.

4: X(M∗INCX) – ***real*** array *Input*

On entry: X$((j-1)$INCX $+ 1)$ must contain the value of the jth variable for the current observation, $j = 1, 2, \ldots, m$.

5: INCX – INTEGER *Input*

On entry: the increment of X. Two situations are commom.

If INCX $= 1$, the data values are to be found in consecutive locations in X, i.e., in a column.

If INCX $=$ LDX, for some positive integer LDX, then the data values are to be found as a row of an array with first dimension LDX.

Constraint: INCX > 0.

6: SW – ***real*** *Input/Output*

On entry: the sum of weights for the previous observations, W_{i-1}.

If SW $= 0.0$, the update procedure is initialised.

If SW $+$ WT $= 0.0$, then all elements of XBAR and C are set to zero.

Constraint: SW ≥ 0.0 and SW $+$ WT ≥ 0.0.

On exit: SW contains the updated sum of weights, W_i.

7: XBAR(M) – ***real*** array *Input/Output*

On entry: XBAR(j) must contain the weighted mean of the jth variable for the previous $(i-1)$ observations, $\bar{x}_j(i-1)$, for $j = 1, 2, \ldots, m$.

On exit: XBAR(j) contains the weighted mean of the jth variable, $\bar{x}_j(i)$. $j = 1, 2, \ldots, m$.

8: C((M∗M+M)/2) – ***real*** array *Input/Output*

On entry: if SW $\neq 0.0$, C must contain the upper triangular part of the matrix of weighted sums of squares and cross-products or weighted sums of squares and cross-products of deviations about the mean. It is stored packed form by column, i.e., the cross-product between the jth and kth variable, $k \geq j$, is stored in C$(k \times (k-1)/2 + j)$.

On exit: the update sums of squares and cross-products stored as on input.

9: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

On exit: IFAIL $= 0$ unless the routine detects an error (see Section 6).

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, for users not familiar with this parameter the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL $= 1$

On entry, M < 1,
or INCX < 1.

IFAIL $= 2$

On entry, SW < 0.0.

IFAIL $= 3$

On entry, (SW + WT) < 0.0, the current weight causes the sum of weights to be less than 0.0.

IFAIL $= 4$

On entry, MEAN $\neq$ 'M' or 'Z'.

7 Accuracy

For a detailed discussion of the accuracy of this method see Chan *et al.* (1982) and West (1979).

8 Further Comments

G02BTF may be used to update the results returned by G02BUF.

G02BWF may be used to calculate the correlation matrix from the matrix of sums of squares and cross-products of deviations about the mean and F06EDF (SSCAL/DSCAL) or F06FDF may be used to scale the matrix to produce a variance-covariance matrix.

9 Example

A program to calculate the means, the required sums of squares and cross-products matrix, and the variance matrix for a set of 3 observations of 3 variables.

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     G02BTF Example Program Text
*     Mark 14 Release.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          INCX, MMAX, MP
      PARAMETER        (INCX=1,MMAX=18,MP=(MMAX*(MMAX+1))/2)
      real             ONE
      PARAMETER        (ONE=1.0e0)
*     .. Local Scalars ..
      real             ALPHA, SW, WT
      INTEGER          I, IFAIL, J, M, MM, N, NPRINT
      CHARACTER        MEAN
*     .. Local Arrays ..
      real             C(MP), V(MP), X(MMAX*INCX), XBAR(MMAX)
*     .. External Subroutines ..
      EXTERNAL         F06FDF, G02BTF, X04CCF
*     .. Intrinsic Functions ..
```

```
      INTRINSIC        MOD
*     .. Executable Statements ..
      WRITE (NOUT,*) 'G02BTF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*,END=40) MEAN, M, N, NPRINT
      SW = 0.0e0
      IF (M.LT.MMAX) THEN
         DO 20 I = 1, N
            READ (NIN,*) WT, (X(J),J=1,M)
            IFAIL = 0
*
*           Calculate the sums of squares and cross-products matrix
            CALL G02BTF(MEAN,M,WT,X,INCX,SW,XBAR,C,IFAIL)
*
            IF (MOD(I,NPRINT).EQ.0 .OR. I.EQ.N) THEN
               WRITE (NOUT,*)
               WRITE (NOUT,*)
     +           '------------------------------------------'
               WRITE (NOUT,99999) 'Observation: ', I, '      Weight = ',
     +           WT
               WRITE (NOUT,*)
     +           '------------------------------------------'
               WRITE (NOUT,*)
               WRITE (NOUT,*) 'Means'
               WRITE (NOUT,99998) (XBAR(J),J=1,M)
               WRITE (NOUT,*)
*              Print the sums of squares and cross products matrix
               CALL X04CCF('Upper','Non-unit',M,C,
     +                     'Sums of squares and cross-products',IFAIL)
               IF (SW.GT.ONE) THEN
*                 Calculate the variance matrix
                  ALPHA = ONE/(SW-ONE)
                  MM = (M*(M+1))/2
                  CALL F06FDF(MM,ALPHA,C,1,V,1)
*                 Print the variance matrix
                  WRITE (NOUT,*)
                  CALL X04CCF('Upper','Non-unit',M,V,'Variance matrix',
     +                        IFAIL)
               END IF
            END IF
   20    CONTINUE
      ELSE
         WRITE (NOUT,99997) 'M is too large. M =', M
      END IF
   40 STOP
*
99999 FORMAT (1X,A,I4,A,F13.4)
99998 FORMAT (1X,4F14.4)
99997 FORMAT (1X,A,I5)
      END
```

9.2 Program Data

```
G02BTF Example Program Data
  'M'  3  3 3
  0.1300   9.1231   3.7011   4.5230
  1.3070   0.9310   0.0900   0.8870
  0.3700   0.0009   0.0099   0.0999
```

9.3 Program Results

```
G02BTF Example Program Results

-------------------------------------------
Observation:    3      Weight =        0.3700
-------------------------------------------

Means
        1.3299        0.3334        0.9874

Sums of squares and cross-products
             1          2          3
1       8.7569     3.6978     4.0707
2                  1.5905     1.6861
3                             1.9297

Variance matrix
             1          2          3
1      10.8512     4.5822     5.0443
2                  1.9709     2.0893
3                             2.3912
```