G02BUF – NAG Fortran Library Routine Document

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

G02BUF calculates the sample means and sums of squares and cross-products, or sums of squares and cross-products of deviations from the mean, in a single pass for a set of data. The data may be weighted.

2 Specification

```
 SUBROUTINE G02BUF(MEAN, WEIGHT, N, M, X, LDX, WT, SW, WMEAN, C,
1                  IFAIL)
 INTEGER           N, M, LDX, IFAIL
 real              X(LDX,M), WT(*), SW, WMEAN(M), C((M*M+M)/2)
 CHARACTER*1       MEAN, WEIGHT
```

3 Description

G02BUF is an adaptation of West's WV2 algorithm, see [2]. This routine calculates the (optionally weighted) sample means and (optionally weighted) sums of squares and cross-products or sums of squares and cross-products of deviations from the (weighted) mean for a sample of n observations on m variables X_j, for $j = 1, 2, \ldots, m$. The algorithm makes a single pass through the data.

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:

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

and

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

The algorithm is initialized 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.

Note that only the upper triangle of the matrix is calculated and returned packed by column.

4 References

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

[2] West D H D (1979) Updating mean and variance estimates: An improved method *Comm. ACM* **22** 532–535

5 Parameters

1: MEAN — CHARACTER*1 *Input*

On entry: indicates whether G02BUF is to calculate sums of squares and cross-products, or sums of squares and cross-products of deviations from the mean.

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

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

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

2: WEIGHT — CHARACTER*1 *Input*

On entry: indicates whether the data is weighted or not.

If WEIGHT = 'U', the calculations are performed on unweighted data.

If WEIGHT = 'W', the calculations are performed on weighted data.

Constraint: WEIGHT = 'W' or 'U'.

3: N — INTEGER *Input*

On entry: the number of observations in the data set, n.

Constraint: N > 1.

4: M — INTEGER *Input*

On entry: the number of variables, m.

Constraint: M > 1.

5: X(LDX,M) — ***real*** array *Input*

On entry: X(i,j) must contain the ith observation on the jth variable, for $i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, m$.

6: LDX — INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which G02BUF is called.

Constraint: LDX $\geq$ N.

7: WT($*$) — ***real*** array *Input*

On entry: the optional weights of each observation.

If WEIGHT = 'W', then W(i) must contain the weight for the ith observation.

If WEIGHT = 'U', then W is not referenced.

Constraint: if WEIGHT = 'W', W$(i) \geq 0.0$, for $i = 1, 2, \ldots, n$.

8: SW — ***real*** *Output*

On exit: the sum of weights.

If WEIGHT = 'U', then SW contains the number of observations, n.

9: WMEAN(M) — ***real*** array *Output*

On exit: the sample means. WMEAN(j) contains the mean for the jth variable.

10: C((M$*$M+M)/2) — ***real*** array *Output*

On exit: the cross-products.

If MEAN = 'M', then C contains the upper triangular part of the matrix of (weighted) sums of squares and cross-products of deviations about the mean.

If MEAN = 'Z', then C contains the upper triangular part of the matrix of (weighted) sums of squares and cross-products.

These are stored packed by columns, i.e., the cross-product between the jth and kth variable $k \geq j$, is stored in C$(k \times (k-1)/2 + j)$.

11: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, −1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

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

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 detected by the routine:

IFAIL = 1

On entry, M < 1,
or N < 1,
or LDX < N.

IFAIL = 2

On entry, MEAN ≠ 'M', or 'Z'.

IFAIL = 3

On entry, WEIGHT ≠ 'W' or 'U'.

IFAIL = 4

On entry, WEIGHT = 'W', and a value of WT < 0.0.

7 Accuracy

For a detailed discussion of the accuracy of this algorithm see Chan *et al.* [1] or West [2].

8 Further Comments

G02BWF may be used to calculate the correlation coefficients from the cross-products of deviations about the mean and F06EDF or F06FDF may be used to scale the cross-products of deviations about the mean to give a variance-covariance matrix.

The means and cross-products produced by G02BUF may be updated by adding or removing observations using G02BTF.

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.

```
*     G02BUF Example Program Text
*     Mark 14 Release.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
```

```
      INTEGER          LDX, MMAX, MP
      PARAMETER        (LDX=12,MMAX=12,MP=(MMAX*(MMAX+1))/2)
      real             ONE
      PARAMETER        (ONE=1.0e0)
*     .. Local Scalars ..
      real             ALPHA, SW
      INTEGER          IFAIL, J, K, M, MM, N
      CHARACTER        MEAN, WEIGHT
*     .. Local Arrays ..
      real             C(MP), V(MP), WMEAN(MMAX), WT(LDX), X(LDX,MMAX)
*     .. External Subroutines ..
      EXTERNAL         F06FDF, G02BUF, X04CCF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'G02BUF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*,END=20) MEAN, WEIGHT, M, N
      IF (M.LE.MMAX .AND. N.LE.LDX) THEN
         READ (NIN,*) (WT(J),J=1,N)
         READ (NIN,*) ((X(J,K),K=1,M),J=1,N)
         IFAIL = 0
*
*        Calculate sums of squares and cross-products matrix
         CALL G02BUF(MEAN,WEIGHT,N,M,X,LDX,WT,SW,WMEAN,C,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Means'
         WRITE (NOUT,99999) (WMEAN(J),J=1,M)
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Weights'
         WRITE (NOUT,99999) (WT(J),J=1,N)
         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
      ELSE
         WRITE (NOUT,99998) 'M or N is too large. M =', M, ', N =', N
      END IF
   20 STOP
*
99999 FORMAT (1X,6F14.4)
99998 FORMAT (1X,A,I6,A,I6)
      END
```

9.2 Program Data

```
G02BUF Example Program Data
 'M'      'W'     3     3
  0.1300   1.3070   0.3700
  9.1231   3.7011   4.5230
  0.9310   0.0900   0.8870
  0.0009   0.0099   0.0999
```

9.3 Program Results

```
G02BUF Example Program Results

Means
        1.3299        0.3334        0.9874

Weights
        0.1300        1.3070        0.3700

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
```