NAG Library Routine Document

G01ABF

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

G01ABF computes the means, standard deviations, corrected sums of squares and products, maximum and minimum values, and the product-moment correlation coefficient for two variables. Unequal weighting may be given.

2 Specification

```
SUBROUTINE G01ABF (N, X1, X2, IWT, WT, RES, IFAIL)
INTEGER                 N, IWT, IFAIL
REAL (KIND=nag_wp) X1(N), X2(N), WT(N), RES(13)
```

3 Description

The data consist of two samples of n observations, denoted by x_i, and y_i, for $i = 1, 2, \ldots, n$, with corresponding weights w_i, for $i = 1, 2, \ldots, n$.

If no specific weighting is given, then each w_i is set to 1.0 in G01ABF.

The quantities calculated are:

(a) The sum of weights,

$$W = \sum_{i=1}^{n} w_i.$$

(b) The means,

$$\bar{x} = \frac{\sum_{i=1}^{n} w_i x_i}{W}, \quad \bar{y} = \frac{\sum_{i=1}^{n} w_i y_i}{W}.$$

(c) The corrected sums of squares and products

$$c_{11} = \sum_{i=1}^{n} w_i (x_i - \bar{x})^2$$

$$c_{21} = c_{12} = \sum_{i=1}^{n} w_i (x_i - \bar{x})(y_i - \bar{y})$$

$$c_{22} = \sum_{i=1}^{n} w_i (y_i - \bar{y})^2.$$

(d) The standard deviations

$$s_j = \sqrt{\frac{c_{jj}}{d}}, \quad \text{where} \quad j = 1, 2 \quad \text{and} \quad d = W - \frac{\sum_{i=1}^{n} w_i^2}{W}.$$

(e) The product-moment correlation coefficient

$$R = \frac{c_{12}}{\sqrt{c_{11}c_{22}}}.$$

(f) The minimum and maximum elements in each of the two samples.

(g) The number of pairs of observations, m, for which $w_i > 0$, i.e., the number of **valid** observations. The quantities in (d) and (e) above will only be computed if $m \geq 2$. All other items are computed if $m \geq 1$.

4 References

None.

5 Parameters

1: N – INTEGER *Input*

On entry: n, the number of pairs of observations.

Constraint: N ≥ 1.

2: X1(N) – REAL (KIND=nag_wp) array *Input*

On entry: the observations from the first sample, x_i, for $i = 1, 2, \ldots, n$.

3: X2(N) – REAL (KIND=nag_wp) array *Input*

On entry: the observations from the second sample, y_i, for $i = 1, 2, \ldots, n$.

4: IWT – INTEGER *Input/Output*

On entry: indicates whether user-supplied weights are provided by you:

IWT $= 1$

Indicates that user-supplied weights are given in the array WT.

IWT $\neq 0$

Indicates that user-supplied weights are not given. In this case the routine assigns the value 1.0 to each element of the weight array, WT.

On exit: is used to indicate the number of valid observations, m; see Section 3(g), above.

5: WT(N) – REAL (KIND=nag_wp) array *Input/Output*

On entry: if weights are being supplied then the elements of WT must contain the weights associated with the observations, w_i, for $i = 1, 2, \ldots, n$.

Constraint: if IWT $= 1$, WT$(i) \geq 0.0$, for $i = 1, 2, \ldots,$ N.

On exit: if IWT $= 1$, the elements of WT are unchanged, otherwise each element of WT will be assigned the value 1.0.

6: RES(13) – REAL (KIND=nag_wp) array *Output*

On exit: the elements of RES contain the following results:

RES(1) mean of the first sample, $\bar{x}$;
RES(2) mean of the second sample, $\bar{y}$;
RES(3) standard deviation of the first sample, s_1;
RES(4) standard deviation of the second sample, s_2;
RES(5) corrected sum of squares of the first sample, c_{11};
RES(6) corrected sum of products of the two samples, c_{12};

RES(7) corrected sum of squares of the second sample, c_{22};
RES(8) product-moment correlation coefficient, R;
RES(9) minimum of the first sample;
RES(10) maximum of the first sample;
RES(11) minimum of the second sample;
RES(12) maximum of the second sample;
RES(13) sum of weights, $\sum_{i=1}^{n} w_i$ (= N, if IWT = 0, on entry).

7: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, −1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

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, if you are 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.**

On exit: IFAIL = 0 unless the routine detects an error or a warning has been flagged (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 or warnings detected by the routine:

IFAIL = 1

On entry, N < 1.

IFAIL = 2

The number of valid cases, m, is 1, hence the standard deviation, 3(d), and the product-moment correlation coefficient, 3(e), cannot be calculated.

IFAIL = 3

The number of valid cases, m, is 0, or at least one of the weights is negative.

IFAIL = −99

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.8 in the Essential Introduction for further information.

IFAIL = −399

Your licence key may have expired or may not have been installed correctly.

See Section 3.7 in the Essential Introduction for further information.

IFAIL = −999

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

7 Accuracy

The method used is believed to be stable.

8 Parallelism and Performance

Not applicable.

9 Further Comments

The time taken by G01ABF increases linearly with n.

10 Example

In the program below, NPROB determines the number of datasets to be analysed. For each analysis, a set of observations and, optionally, weights, is read and printed. After calling G01ABF, all the calculated quantities are printed. In the example, there is one set of data, with 29 (unweighted) pairs of observations.

10.1 Program Text

```
    Program g01abfe

!     G01ABF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: g01abf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, iwt, n
!     .. Local Arrays ..
      Real (Kind=nag_wp)               :: res(13)
      Real (Kind=nag_wp), Allocatable  :: wt(:), wtin(:), x1(:), x2(:)
!     .. Executable Statements ..
      Write (nout,*) 'G01ABF Example Program Results'
      Write (nout,*)

!     Skip heading in data file
      Read (nin,*)

!     Read in the problem size
      Read (nin,*) n, iwt

      Allocate (wt(n),wtin(n),x1(n),x2(n))

!     Read in data
      Read (nin,*)(x1(i),x2(i),i=1,n)
      If (iwt==1) Then
        Read (nin,*) wtin(1:n)
        wt(1:n) = wtin(1:n)
      End If

!     Display data
      Write (nout,99999) 'Number of cases', n
      Write (nout,*) 'Data as input -'
      Write (nout,*) '         Var  1     Var  2     Var  1    &
        & Var  2     Var  1     Var  2'

      Write (nout,99995)(x1(i),x2(i),i=1,n)
      If (iwt==1) Then
        Write (nout,*) 'Weights as input -'
        Write (nout,99994) wtin(1:n)
      End If
      Write (nout,*)

!     Calculate summary statistics
```

```
      ifail = -1
      Call g01abf(n,x1,x2,iwt,wt,res,ifail)
      If (ifail/=0) Then
        If (ifail/=2) Then
          Go To 100
        End If
      End If

!     Display results
      Write (nout,99999) 'No. of valid cases', iwt
      Write (nout,99993) 'Variable 1', 'Variable 2'
      Write (nout,99998) 'Mean    ', res(1), res(2)
      Write (nout,99997) 'Corr SSP', res(5), res(6), res(7)
      Write (nout,99998) 'Minimum ', res(9), res(11)
      Write (nout,99998) 'Maximum ', res(10), res(12)
      Write (nout,99998) 'Sum of weights         ', res(13)
      If (ifail==0) Then
        Write (nout,99998) 'Std devn', res(3), res(4)
        Write (nout,99996) 'Correln ', res(8)
      Else
        Write (nout,*) 'Std devn and Correln not defined'
      End If

100   Continue

99999 Format (1X,A,I5)
99998 Format (1X,A,F15.1,F30.1)
99997 Format (1X,A,3E15.5)
99996 Format (1X,A,F30.4)
99995 Format (5X,6F11.1)
99994 Format (13X,F9.3)
99993 Format (13X,A,20X,A)
    End Program g01abfe
```

10.2 Program Data

```
G01ABF Example Program Data
 29  0
    350     47    550     95    380    211    510    122   1270    530
    300     38   2630    278    810    309    140     75    450     43
   2280    407    250    142    540     89    720    159     90     35
    480    103    180     78   3160    969    220    120    860    333
    300     73   1460    147    400     30    620    100    120     55
    780    145    230    101   1070    468    160     86
```

10.3 Program Results

```
 G01ABF Example Program Results

 Number of cases   29
 Data as input -
          Var  1     Var  2     Var  1     Var  2     Var  1     Var  2
           350.0       47.0      550.0       95.0      380.0      211.0
           510.0      122.0     1270.0      530.0      300.0       38.0
          2630.0      278.0      810.0      309.0      140.0       75.0
           450.0       43.0     2280.0      407.0      250.0      142.0
           540.0       89.0      720.0      159.0       90.0       35.0
           480.0      103.0      180.0       78.0     3160.0      969.0
           220.0      120.0      860.0      333.0      300.0       73.0
          1460.0      147.0      400.0       30.0      620.0      100.0
           120.0       55.0      780.0      145.0      230.0      101.0
          1070.0      468.0      160.0       86.0

 No. of valid cases   29
             Variable 1                    Variable 2
 Mean             734.8                         185.8
 Corr SSP    0.16396E+08    0.34830E+07    0.11319E+07
```

```
Minimum            90.0                          30.0
Maximum          3160.0                         969.0
Sum of weights                29.0
Std devn          765.2                         201.1
Correln                      0.8085
```