# NAG Library Routine Document

# G03EAF

**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

G03EAF computes a distance (dissimilarity) matrix.

## 2 Specification

```
SUBROUTINE G03EAF (UPDATE, DIST, SCAL, N, M, X, LDX, ISX, S, D, IFAIL)

INTEGER             N, M, LDX, ISX(M), IFAIL
REAL (KIND=nag_wp) X(LDX,M), S(M), D(N*(N-1)/2)
CHARACTER(1)        UPDATE, DIST, SCAL
```

## 3 Description

Given $n$ objects, a distance or dissimilarity matrix is a symmetric matrix with zero diagonal elements such that the $ij$th element represents how far apart or how dissimilar the $i$th and $j$th objects are.

Let $X$ be an $n$ by $p$ data matrix of observations of $p$ variables on $n$ objects, then the distance between object $j$ and object $k$, $d_{jk}$, can be defined as:

$$d_{jk} = \left\{ \sum_{i=1}^{p} D\left(x_{ji}/s_i, x_{ki}/s_i\right) \right\}^{\alpha},$$

where $x_{ji}$ and $x_{ki}$ are the $ji$th and $ki$th elements of $X$, $s_i$ is a standardization for the $i$th variable and $D(u,v)$ is a suitable function. Three functions are provided in G03EAF.

(a) Euclidean distance: $D(u,v) = (u-v)^2$ and $\alpha = \frac{1}{2}$.

(b) Euclidean squared distance: $D(u,v) = (u-v)^2$ and $\alpha = 1$.

(c) Absolute distance (city block metric): $D(u,v) = |u-v|$ and $\alpha = 1$.

Three standardizations are available.

(a) Standard deviation: $s_i = \sqrt{\sum_{j=1}^{n} \left(x_{ji} - \bar{x}\right)^2 / (n-1)}$

(b) Range: $s_i = \max(x_{1i}, x_{2i}, \ldots, x_{ni}) - \min(x_{1i}, x_{2i}, \ldots, x_{ni})$

(c) User-supplied values of $s_i$.

In addition to the above distances there are a large number of other dissimilarity measures, particularly for dichotomous variables (see Krzanowski (1990) and Everitt (1974)). For the dichotomous case these measures are simple to compute and can, if suitable scaling is used, be combined with the distances computed by G03EAF using the updating option.

Dissimilarity measures for variables can be based on the correlation coefficient for continuous variables and contingency table statistics for dichotomous data, see chapters G02 and G11 respectively.

G03EAF returns the strictly lower triangle of the distance matrix.

## 4 References

Everitt B S (1974) *Cluster Analysis* Heinemann

Krzanowski W J (1990) *Principles of Multivariate Analysis* Oxford University Press

## 5 Parameters

1: UPDATE – CHARACTER(1) *Input*

*On entry*: indicates whether or not an existing matrix is to be updated.

UPDATE = 'U'
The matrix $D$ is updated and distances are added to $D$.

UPDATE = 'I'
The matrix $D$ is initialized to zero before the distances are added to $D$.

*Constraint*: UPDATE = 'U' or 'I'.

2: DIST – CHARACTER(1) *Input*

*On entry*: indicates which type of distances are computed.

DIST = 'A'
Absolute distances.

DIST = 'E'
Euclidean distances.

DIST = 'S'
Euclidean squared distances.

*Constraint*: DIST = 'A', 'E' or 'S'.

3: SCAL – CHARACTER(1) *Input*

*On entry*: indicates the standardization of the variables to be used.

SCAL = 'S'
Standard deviation.

SCAL = 'R'
Range.

SCAL = 'G'
Standardizations given in array S.

SCAL = 'U'
Unscaled.

*Constraint*: SCAL = 'S', 'R', 'G' or 'U'.

4: N – INTEGER *Input*

*On entry*: $n$, the number of observations.

*Constraint*: N $\geq 2$.

5: M – INTEGER *Input*

*On entry*: the total number of variables in array X.

*Constraint*: M $> 0$.

6: X(LDX,M) – REAL (KIND=nag_wp) array *Input*

*On entry*: X$(i,j)$ must contain the value of the $j$th variable for the $i$th object, for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots,$ M.

7: LDX – INTEGER *Input*

*On entry*: the first dimension of the array X as declared in the (sub)program from which G03EAF is called.

*Constraint*: LDX $\geq$ N.

8: ISX(M) – INTEGER array *Input*

*On entry*: ISX($j$) indicates whether or not the $j$th variable in X is to be included in the distance computations.

If ISX($j$) $> 0$ the $j$th variable is included, for $j = 1, 2, \ldots, \mathrm{M}$; otherwise it is not referenced.

*Constraint*: ISX($j$) $> 0$ for at least one $j$, for $j = 1, 2, \ldots, \mathrm{M}$.

9: S(M) – REAL (KIND=nag_wp) array *Input/Output*

*On entry*: if SCAL = 'G' and ISX($j$) $> 0$ then S($j$) must contain the scaling for variable $j$, for $j = 1, 2, \ldots, \mathrm{M}$.

*Constraint*: if SCAL = 'G' and ISX($j$) $> 0$, S($j$) $> 0.0$, for $j = 1, 2, \ldots, \mathrm{M}$.

*On exit*: if SCAL = 'S' and ISX($j$) $> 0$ then S($j$) contains the standard deviation of the variable in the $j$th column of X.

If SCAL = 'R' and ISX($j$) $> 0$, S($j$) contains the range of the variable in the $j$th column of X.

If SCAL = 'U' and ISX($j$) $> 0$, S($j$) $= 1.0$.

If SCAL = 'G', S is unchanged.

10: D(N $\times$ (N $-$ 1)/2) – REAL (KIND=nag_wp) array *Input/Output*

*On entry*: if UPDATE = 'U', D must contain the strictly lower triangle of the distance matrix $D$ to be updated. $D$ must be stored packed by rows, i.e., D($(i-1)(i-2)/2 + j$), $i > j$ must contain $d_{ij}$.

If UPDATE = 'I', D need not be set.

*Constraint*: if UPDATE = 'U', D($j$) $\geq 0.0$, for $j = 1, 2, \ldots, n(n-1)/2$.

*On exit*: the strictly lower triangle of the distance matrix $D$ stored packed by rows, i.e., $d_{ij}$ is contained in D($(i-1)(i-2)/2 + j$), $i > j$.

11: 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 $< 2$,
or LDX $<$ N,
or M $\leq 0$,
or UPDATE $\neq$ 'I' or 'U',
or DIST $\neq$ 'A', 'E' or 'S',
or SCAL $\neq$ 'S', 'R', 'G' or 'U'.

IFAIL $= 2$

On entry, ISX$(j) \leq 0$, for $j = 1, 2, \ldots,$ M,
or UPDATE $=$ 'U' and D$(j) < 0.0$, for some $j = 1, 2, \ldots, n(n-1)/2$,
or SCAL $=$ 'S' or 'R' and X$(i, j) =$ X$(i+1, j)$ for $i = 1, 2, \ldots, n-1$, for some $j$ with ISX$(i) > 0$.
or S$(j) \leq 0.0$ for some $j$ when SCAL $=$ 'G' and ISX$(j) > 0$.

## 7 Accuracy

The computations are believed to be stable.

## 8 Further Comments

G03ECF can be used to perform cluster analysis on the computed distance matrix.

## 9 Example

A data matrix of five observations and three variables is read in and a distance matrix is calculated from variables 2 and 3 using squared Euclidean distance with no scaling. This matrix is then printed.

### 9.1 Program Text

```
    Program g03eafe

!     G03EAF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: g03eaf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, ld, ldx, lj, m, n, uj
      Character (1)                    :: dist, scal, update
      Character (80)                   :: fmt
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: d(:), s(:), x(:,:)
      Integer, Allocatable             :: isx(:)
!     .. Executable Statements ..
      Write (nout,*) 'G03EAF Example Program Results'
      Write (nout,*)

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

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

!     Read in information on the type of distance matrix to use
      Read (nin,*) update, dist, scal

      ldx = n
      ld = n*(n-1)/2
      Allocate (x(ldx,m),isx(m),s(m),d(ld))

!     Read in the data used to construct distance matrix
      Read (nin,*)(x(i,1:m),i=1,n)

!     Read in variable inclusion flags
      Read (nin,*) isx(1:m)

!     Read in scaling
      If (scal=='G' .Or. scal=='g') Then
        Read (nin,*) s(1:m)
      End If

!     Compute the distance matrix
      ifail = 0
      Call g03eaf(update,dist,scal,n,m,x,ldx,isx,s,d,ifail)

!     Display results
      Write (nout,*) ' Distance Matrix'
      Write (nout,*)
      Write (fmt,99999) '(3X,', n - 1, 'I8)'
      Write (nout,fmt)(i,i=1,n-1)
      Write (nout,*)
      Write (fmt,99999) '(1X,I2,2X,', n - 1, '(3X,F5.2))'
      Do i = 2, n
        lj = (i-1)*(i-2)/2 + 1
        uj = i*(i-1)/2
        Write (nout,fmt) i, d(lj:uj)
      End Do

99999 Format (A,I0,A)
    End Program g03eafe
```

## 9.2 Program Data

```
G03EAF Example Program Data
5 3          : N,M
'I' 'S' 'U'  : UPDATE,DIST,SCAL
1.0 1.0 1.0
2.0 1.0 2.0
3.0 6.0 3.0
4.0 8.0 2.0
5.0 8.0 0.0  : End of X
 0   1   1   : ISX
```

## 9.3 Program Results

```
 G03EAF Example Program Results

  Distance Matrix

           1       2       3       4

  2      1.00
  3     29.00   26.00
  4     50.00   49.00    5.00
  5     50.00   53.00   13.00    4.00
```