NAG Library Routine Document

G03DBF

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

G03DBF computes Mahalanobis squared distances for group or pooled variance-covariance matrices. It is intended for use after G03DAF.

2 Specification

```
SUBROUTINE G03DBF (EQUAL, MODE, NVAR, NG, GMN, LDGMN, GC, NOBS, M, ISX,     &
                   X, LDX, D, LDD, WK, IFAIL)

INTEGER             NVAR, NG, LDGMN, NOBS, M, ISX(*), LDX, LDD, IFAIL
REAL (KIND=nag_wp) GMN(LDGMN,NVAR), GC((NG+1)*NVAR*(NVAR+1)/2),            &
                   X(LDX,*), D(LDD,NG), WK(2*NVAR)
CHARACTER(1)        EQUAL, MODE
```

3 Description

Consider p variables observed on n_g populations or groups. Let $\bar{x}_j$ be the sample mean and S_j the within-group variance-covariance matrix for the jth group and let x_k be the kth sample point in a dataset. A measure of the distance of the point from the jth population or group is given by the Mahalanobis distance, D_{kj}:

$$D_{kj}^2 = \left(x_k - \bar{x}_j\right)^{\mathrm{T}} S_j^{-1} \left(x_k - \bar{x}_j\right).$$

If the pooled estimated of the variance-covariance matrix S is used rather than the within-group variance-covariance matrices, then the distance is:

$$D_{kj}^2 = \left(x_k - \bar{x}_j\right)^{\mathrm{T}} S^{-1} \left(x_k - \bar{x}_j\right).$$

Instead of using the variance-covariance matrices S and S_j, G03DBF uses the upper triangular matrices R and R_j supplied by G03DAF such that $S = R^{\mathrm{T}}R$ and $S_j = R_j^{\mathrm{T}}R_j$. D_{kj}^2 can then be calculated as $z^{\mathrm{T}}z$ where $R_j z = \left(x_k - \bar{x}_j\right)$ or $Rz = \left(x_k - \bar{x}_j\right)$ as appropriate.

A particular case is when the distance between the group or population means is to be estimated. The Mahalanobis squared distance between the ith and jth groups is:

$$D_{ij}^2 = \left(\bar{x}_i - \bar{x}_j\right)^{\mathrm{T}} S_j^{-1} \left(\bar{x}_i - \bar{x}_j\right)$$

or

$$D_{ij}^2 = \left(\bar{x}_i - \bar{x}_j\right)^{\mathrm{T}} S^{-1} \left(\bar{x}_i - \bar{x}_j\right).$$

Note: $D_{jj}^2 = 0$ and that in the case when the pooled variance-covariance matrix is used $D_{ij}^2 = D_{ji}^2$ so in this case only the lower triangular values of D_{ij}^2, $i > j$, are computed.

4 References

Aitchison J and Dunsmore I R (1975) *Statistical Prediction Analysis* Cambridge

Kendall M G and Stuart A (1976) *The Advanced Theory of Statistics (Volume 3)* (3rd Edition) Griffin

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

5 Arguments

1: EQUAL – CHARACTER(1) *Input*

On entry: indicates whether or not the within-group variance-covariance matrices are assumed to be equal and the pooled variance-covariance matrix used.

EQUAL = 'E'
The within-group variance-covariance matrices are assumed equal and the matrix R stored in the first $p(p+1)/2$ elements of GC is used.

EQUAL = 'U'
The within-group variance-covariance matrices are assumed to be unequal and the matrices R_j, for $j = 1, 2, \ldots, n_g$, stored in the remainder of GC are used.

Constraint: EQUAL = 'E' or 'U'.

2: MODE – CHARACTER(1) *Input*

On entry: indicates whether distances from sample points are to be calculated or distances between the group means.

MODE = 'S'
The distances between the sample points given in X and the group means are calculated.

MODE = 'M'
The distances between the group means will be calculated.

Constraint: MODE = 'M' or 'S'.

3: NVAR – INTEGER *Input*

On entry: p, the number of variables in the variance-covariance matrices as specified to G03DAF.

Constraint: NVAR ≥ 1.

4: NG – INTEGER *Input*

On entry: the number of groups, n_g.

Constraint: NG ≥ 2.

5: GMN(LDGMN, NVAR) – REAL (KIND=nag_wp) array *Input*

On entry: the jth row of GMN contains the means of the p selected variables for the jth group, for $j = 1, 2, \ldots, n_g$. These are returned by G03DAF.

6: LDGMN – INTEGER *Input*

On entry: the first dimension of the array GMN as declared in the (sub)program from which G03DBF is called.

Constraint: LDGMN $\geq$ NG.

7: GC((NG + 1) × NVAR × (NVAR + 1)/2) – REAL (KIND=nag_wp) array *Input*

On entry: the first $p(p+1)/2$ elements of GC should contain the upper triangular matrix R and the next n_g blocks of $p(p+1)/2$ elements should contain the upper triangular matrices R_j. All matrices must be stored packed by column. These matrices are returned by G03DAF. If EQUAL = 'E' only the first $p(p+1)/2$ elements are referenced, if EQUAL = 'U' only the elements $p(p+1)/2 + 1$ to $\big(n_g + 1\big)p(p+1)/2$ are referenced.

Constraints:

if EQUAL = 'E', $R \neq 0.0$;
if EQUAL = 'U', the diagonal elements of the $R_j \neq 0.0$, for $j = 1, 2, \ldots,$ NG.

8: NOBS – INTEGER *Input*

On entry: if MODE = 'S', the number of sample points in X for which distances are to be calculated.

If MODE = 'M', NOBS is not referenced.

Constraint: if NOBS ≥ 1, MODE = 'S'.

9: M – INTEGER *Input*

On entry: if MODE = 'S', the number of variables in the data array X.

If MODE = 'M', M is not referenced.

Constraint: if M $\geq$ NVAR, MODE = 'S'.

10: ISX($*$) – INTEGER array *Input*

Note: the dimension of the array ISX must be at least $\max(1, \mathrm{M})$.

On entry: if MODE = 'S', ISX(l) indicates if the lth variable in X is to be included in the distance calculations. If ISX(l) > 0 the lth variable is included, for $l = 1, 2, \ldots, \mathrm{M}$; otherwise the lth variable is not referenced.

If MODE = 'M', ISX is not referenced.

Constraint: if MODE = 'S', ISX(l) > 0 for NVAR values of l.

11: X(LDX, $*$) – REAL (KIND=nag_wp) array *Input*

Note: the second dimension of the array X must be at least $\max(1, \mathrm{M})$.

On entry: if MODE = 'S' the kth row of X must contain x_k. That is X(k, l) must contain the kth sample value for the lth variable, for $k = 1, 2, \ldots,$ NOBS and $l = 1, 2, \ldots, \mathrm{M}$. Otherwise X is not referenced.

12: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which G03DBF is called.

Constraints:

if MODE = 'S', LDX $\geq$ NOBS;
otherwise 1.

13: D(LDD, NG) – REAL (KIND=nag_wp) array *Output*

On exit: the squared distances.

If MODE = 'S', D(k, j) contains the squared distance of the kth sample point from the jth group mean, D_{kj}^2, for $k = 1, 2, \ldots,$ NOBS and $j = 1, 2, \ldots, n_g$.

If MODE = 'M' and EQUAL = 'U', D(i, j) contains the squared distance between the ith mean and the jth mean, D_{ij}^2, for $i = 1, 2, \ldots, n_g$ and $j = 1, 2, \ldots, i-1, i+1, \ldots, n_g$. The elements D($i, i$) are not referenced, for $i = 1, 2, \ldots, n_g$.

If MODE = 'M' and EQUAL = 'E', D(i, j) contains the squared distance between the ith mean and the jth mean, D_{ij}^2, for $i = 1, 2, \ldots, n_g$ and $j = 1, 2, \ldots, i-1$. Since $D_{ij} = D_{ji}$ the elements D(i, j) are not referenced, for $i = 1, 2, \ldots, n_g$ and $j = i+1, \ldots, n_g$.

14: LDD – INTEGER *Input*

On entry: the first dimension of the array D as declared in the (sub)program from which G03DBF is called.

Constraints:

if MODE = 'S', LDD $\geq$ NOBS;
if MODE = 'M', LDD $\geq$ NG.

15: WK($2 \times$ NVAR) – REAL (KIND=nag_wp) array *Workspace*

16: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation 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 argument, 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, NVAR $<$ 1,
or NG $<$ 2,
or LDGMN $<$ NG,
or MODE = 'S' and NOBS $<$ 1,
or MODE = 'S' and M $<$ NVAR,
or MODE = 'S' and LDX $<$ NOBS,
or MODE = 'S' and LDD $<$ NOBS,
or MODE = 'M' and LDD $<$ NG,
or EQUAL $\neq$ 'E' or 'U',
or MODE $\neq$ 'M' or 'S'.

IFAIL = 2

On entry, MODE = 'S' and the number of variables indicated by ISX is not equal to NVAR,
or EQUAL = 'E' and a diagonal element of R is zero,
or EQUAL = 'U' and a diagonal element of R_j for some j is zero.

IFAIL = -99

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

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL = -399

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

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL = -999

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

The accuracy will depend upon the accuracy of the input R or R_j matrices.

8 Parallelism and Performance

G03DBF makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

If the distances are to be used for discrimination, see also G03DCF.

10 Example

The data, taken from Aitchison and Dunsmore (1975), is concerned with the diagnosis of three ‘types’ of Cushing's syndrome. The variables are the logarithms of the urinary excretion rates (mg/24hr) of two steroid metabolites. Observations for a total of 21 patients are input and the group means and R matrices are computed by G03DAF. A further six observations of unknown type are input, and the distances from the group means of the 21 patients of known type are computed under the assumption that the within-group variance-covariance matrices are not equal. These results are printed and indicate that the first four are close to one of the groups while observations 5 and 6 are some distance from any group.

10.1 Program Text

```
    Program g03dbfe

!     G03DBF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: g03daf, g03dbf, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: df, sig, stat
      Integer                          :: i, ifail, ldd, ldgmn, ldox, ldx,     &
                                          lgc, lwk, lwt, m, n, ng, nobs, nvar
      Character (1)                    :: equal, mode, weight
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: d(:,:), det(:), gc(:), gmn(:,:),     &
                                          ox(:,:), wk(:), wt(:), x(:,:)
      Integer, Allocatable             :: ing(:), isx(:), iwk(:), nig(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: count, max
!     .. Executable Statements ..
      Write (nout,*) 'G03DBF Example Program Results'
      Write (nout,*)
      Flush (nout)

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

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

```
      If (weight=='W' .Or. weight=='w') Then
        lwt = n
      Else
        lwt = 0
      End If
      ldox = n
      Allocate (ox(ldox,m),ing(n),wt(lwt),isx(m))

!     Read in original data
      If (lwt>0) Then
        Read (nin,*)(ox(i,1:m),ing(i),wt(i),i=1,n)
      Else
        Read (nin,*)(ox(i,1:m),ing(i),i=1,n)
      End If

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

!     Calculate NVAR
      nvar = count(isx(1:m)==1)

      ldgmn = ng
      lgc = (ng+1)*nvar*(nvar+1)/2
      lwk = max(n*(nvar+1),2*nvar)
      Allocate (nig(ng),gmn(ldgmn,nvar),det(ng),gc(lgc),wk(lwk),iwk(ng))

!     Compute covariance matrix
      ifail = 0
      Call g03daf(weight,n,m,ox,ldox,isx,nvar,ing,ng,wt,nig,gmn,ldgmn,det,gc,  &
        stat,df,sig,wk,iwk,ifail)

!     Read in size data from which to compute distances
      Read (nin,*) mode, equal

      If (mode=='S' .Or. mode=='s') Then
        Read (nin,*) nobs
        ldd = nobs
      Else
        nobs = 0
        ldd = ng
      End If

      ldx = nobs
      Allocate (x(ldx,m),d(ldd,ng))

!     Read in data from which to compute distances
      If (nobs>0) Then
        Read (nin,*)(x(i,1:m),i=1,nobs)
      End If

!     Compute distances
      ifail = 0
      Call g03dbf(equal,mode,nvar,ng,gmn,ldgmn,gc,nobs,m,isx,x,ldx,d,ldd,wk,   &
        ifail)

!     Display results
      ifail = 0
      Call x04caf('General',' ',nobs,ng,d,ldd,'Distances',ifail)

    End Program g03dbfe
```

10.2 Program Data

```
G03DBF Example Program Data
  21 2 3 'U'             : N,M,NG,WEIGHT
  1.1314    2.4596    1
  1.0986    0.2624    1
  0.6419   -2.3026    1
  1.3350   -3.2189    1
```

```
 1.4110    0.0953    1
 0.6419   -0.9163    1
 2.1163    0.0000    2
 1.3350   -1.6094    2
 1.3610   -0.5108    2
 2.0541    0.1823    2
 2.2083   -0.5108    2
 2.7344    1.2809    2
 2.0412    0.4700    2
 1.8718   -0.9163    2
 1.7405   -0.9163    2
 2.6101    0.4700    2
 2.3224    1.8563    3
 2.2192    2.0669    3
 2.2618    1.1314    3
 3.9853    0.9163    3
 2.7600    2.0281    3  : End of X,ING (G03EAF)
  1         1           : ISX
  'S' 'U'               : MODE,EQUAL
  6                     : NOBS
 1.6292   -0.9163
 2.5572    1.6094
 2.5649   -0.2231
 0.9555   -2.3026
 3.4012   -2.3026
 3.0204   -0.2231       : End of X
```

10.3 Program Results

```
G03DBF Example Program Results

Distances
            1          2          3
1      3.3393     0.7521    50.9283
2     20.7771     5.6559     0.0597
3     21.3631     4.8411    19.4978
4      0.7184     6.2803   124.7323
5     55.0003    88.8604    71.7852
6     36.1703    15.7849    15.7489
```