NAG Library Routine Document

G02DDF

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

G02DDF calculates the regression parameters for a general linear regression model. It is intended to be called after G02DCF, G02DEF or G02DFF.

2 Specification

```
SUBROUTINE G02DDF (N, IP, Q, LDQ, RSS, IDF, B, SE, COV, SVD, IRANK, P,     &
                   TOL, WK, IFAIL)

INTEGER            N, IP, LDQ, IDF, IRANK, IFAIL
REAL (KIND=nag_wp) Q(LDQ,IP+1), RSS, B(IP), SE(IP), COV(IP*(IP+1)/2),      &
                   P(IP*IP+2*IP), TOL, WK(IP*IP+(IP-1)*5)
LOGICAL            SVD
```

3 Description

A general linear regression model fitted by G02DAF may be adjusted by adding or deleting an observation using G02DCF, adding a new independent variable using G02DEF or deleting an existing independent variable using G02DFF. Alternatively a model may be constructed by a forward selection procedure using G02EEF. These routines compute the vector c and the upper triangular matrix R. G02DDF takes these basic results and computes the regression coefficients, $\hat{\beta}$, their standard errors and their variance-covariance matrix.

If R is of full rank, then $\hat{\beta}$ is the solution to

$$R\hat{\beta} = c_1,$$

where c_1 is the first p elements of c.

If R is not of full rank a solution is obtained by means of a singular value decomposition (SVD) of R,

$$R = Q_* \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} P^{\mathrm{T}},$$

where D is a k by k diagonal matrix with nonzero diagonal elements, k being the rank of R, and Q_* and P are p by p orthogonal matrices. This gives the solution

$$\hat{\beta} = P_1 D^{-1} Q_{*_1}^{\mathrm{T}} c_1.$$

P_1 being the first k columns of P, i.e., $P = (P_1 P_0)$, and Q_{*_1} being the first k columns of Q_*.

Details of the SVD are made available in the form of the matrix P^*:

$$P^* = \begin{pmatrix} D^{-1} P_1^{\mathrm{T}} \\ P_0^{\mathrm{T}} \end{pmatrix}.$$

This will be only one of the possible solutions. Other estimates may be obtained by applying constraints to the parameters. These solutions can be obtained by calling G02DKF after calling G02DDF. Only certain linear combinations of the parameters will have unique estimates; these are known as estimable functions. These can be estimated using G02DNF.

The residual sum of squares required to calculate the standard errors and the variance-covariance matrix can either be input or can be calculated if additional information on c for the whole sample is provided.

4 References

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

Hammarling S (1985) The singular value decomposition in multivariate statistics *SIGNUM Newsl.* **20(3)** 2–25

Searle S R (1971) *Linear Models* Wiley

5 Parameters

1: N – INTEGER *Input*

On entry: the number of observations.

Constraint: $\mathrm{N} \geq 1$.

2: IP – INTEGER *Input*

On entry: p, the number of terms in the regression model.

Constraint: $\mathrm{IP} \geq 1$.

3: Q(LDQ, IP + 1) – REAL (KIND=nag_wp) array *Input*

On entry: must be the array Q as output by G02DCF, G02DEF, G02DFF or G02EEF. If on entry $\mathrm{RSS} \leq 0.0$ then all N elements of c are needed. This is provided by routines G02DEF, G02DFF or G02EEF.

4: LDQ – INTEGER *Input*

On entry: the first dimension of the array Q as declared in the (sub)program from which G02DDF is called.

Constraints:

if $\mathrm{RSS} \leq 0.0$, $\mathrm{LDQ} \geq \mathrm{N}$;
otherwise $\mathrm{LDQ} \geq \mathrm{IP}$.

5: RSS – REAL (KIND=nag_wp) *Input/Output*

On entry: either the residual sum of squares or a value less than or equal to 0.0 to indicate that the residual sum of squares is to be calculated by the routine.

On exit: if $\mathrm{RSS} \leq 0.0$ on entry, then on exit RSS will contain the residual sum of squares as calculated by G02DDF.

If RSS was positive on entry, it will be unchanged.

6: IDF – INTEGER *Output*

On exit: the degrees of freedom associated with the residual sum of squares.

7: B(IP) – REAL (KIND=nag_wp) array *Output*

On exit: the estimates of the p parameters, $\hat{\beta}$.

8: SE(IP) – REAL (KIND=nag_wp) array *Output*

On exit: the standard errors of the p parameters given in B.

9: COV(IP × (IP + 1)/2) – REAL (KIND=nag_wp) array *Output*

On exit: the upper triangular part of the variance-covariance matrix of the p parameter estimates given in B. They are stored packed by column, i.e., the covariance between the parameter estimate

given in B(i) and the parameter estimate given in B(j), $j \geq i$, is stored in COV($j \times (j-1)/2 + i$).

10: SVD – LOGICAL *Output*

On exit: if a singular value decomposition has been performed, SVD = .TRUE., otherwise SVD = .FALSE..

11: IRANK – INTEGER *Output*

On exit: the rank of the independent variables.

If SVD = .FALSE., IRANK = IP.

If SVD = .TRUE., IRANK is an estimate of the rank of the independent variables.

IRANK is calculated as the number of singular values greater than TOL $\times$ (largest singular value). It is possible for the SVD to be carried out but IRANK to be returned as IP.

12: P(IP $\times$ IP + 2 $\times$ IP) – REAL (KIND=nag_wp) array *Output*

On exit: contains details of the singular value decomposition if used.

If SVD = .FALSE., P is not referenced.

If SVD = .TRUE., the first IP elements of P will not be referenced, the next IP values contain the singular values. The following IP $\times$ IP values contain the matrix P^* stored by columns.

13: TOL – REAL (KIND=nag_wp) *Input*

On entry: the value of TOL is used to decide if the independent variables are of full rank and, if not, what is the rank of the independent variables. The smaller the value of TOL the stricter the criterion for selecting the singular value decomposition. If TOL = 0.0, the singular value decomposition will never be used, this may cause run time errors or inaccuracies if the independent variables are not of full rank.

Suggested value: TOL = 0.000001.

Constraint: TOL $\geq$ 0.0.

14: WK(IP $\times$ IP + (IP − 1) $\times$ 5) – REAL (KIND=nag_wp) array *Workspace*

15: 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,
or IP < 1,
or LDQ $<$ IP,
or LDQ $<$ N,
or TOL < 0.0.

IFAIL $= 2$

The degrees of freedom for error are less than or equal to 0. In this case the estimates of β are returned but not the standard errors or covariances.

IFAIL $= 3$

The singular value decomposition, if used, has failed to converge, see F02WUF. This is an unlikely error exit.

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 accuracy of the results will depend on the accuracy of the input R matrix, which may lose accuracy if a large number of observations or variables have been dropped.

8 Parallelism and Performance

G02DDF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

G02DDF 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

None.

10 Example

A dataset consisting of 12 observations and four independent variables is input and a regression model fitted by calls to G02DEF. The parameters are then calculated by G02DDF and the results printed.

10.1 Program Text

```
    Program g02ddfe

!     G02DDF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: g02ddf, g02def, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: rss, tol
      Integer                          :: i, idf, ifail, ip, irank, ldq, lwt,  &
                                          m, n
      Logical                          :: svd
      Character (1)                    :: weight
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: b(:), cov(:), p(:), q(:,:), se(:),   &
                                          wk(:), wt(:), x(:,:)
!     .. Executable Statements ..
      Write (nout,*) 'G02DDF Example Program Results'
      Write (nout,*)

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

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

      If (weight=='W' .Or. weight=='w') Then
        lwt = n
      Else
        lwt = 0
      End If
      ldq = n
      Allocate (b(m),cov(m*(m+1)/2),p(m*(m+2)),q(ldq,m+1),se(m),wk(m*m+5*m),wt &
        (n),x(n,m))

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

!     Use suggested value for tolerance
      tol = 0.000001E0_nag_wp

!     Fit general linear regression model, adding each variable in turn
      ip = 0
      Do i = 1, m
        ifail = -1
        Call g02def(weight,n,ip,q,ldq,p,wt,x(1,i),rss,tol,ifail)
        If (ifail==0) Then
          ip = ip + 1
        Else If (ifail==3) Then
          Write (nout,99996) ' * Variable ', ip, &
            ' is linear combination of previous columns'
          Write (nout,99996) '   so it has not been added'
        Else
```

```
          Go To 100
        End If
      End Do

!     Get G02DDF to calculate RSS
      rss = 0.0E0_nag_wp

!     Calculate parameter estimates, RSS etc
      ifail = 0
      Call g02ddf(n,ip,q,ldq,rss,idf,b,se,cov,svd,irank,p,tol,wk,ifail)

!     Display results
      If (svd) Then
        Write (nout,*) 'Model not of full rank'
        Write (nout,*)
      End If
      Write (nout,99999) 'Residual sum of squares = ', rss
      Write (nout,99998) 'Degrees of freedom = ', idf
      Write (nout,*)
      Write (nout,*) 'Variable   Parameter estimate   Standard error'
      Write (nout,*)
      Write (nout,99997)(i,b(i),se(i),i=1,ip)

100   Continue

99999 Format (1X,A,E12.4)
99998 Format (1X,A,I4)
99997 Format (1X,I6,2E20.4)
99996 Format (1X,A,I0,A)
    End Program g02ddfe
```

10.2 Program Data

```
G02DDF Example Program Data
 12 4 'U'
1.0 0.0 0.0 0.0 33.63
0.0 0.0 0.0 1.0 39.62
0.0 1.0 0.0 0.0 38.18
0.0 0.0 1.0 0.0 41.46
0.0 0.0 0.0 1.0 38.02
0.0 1.0 0.0 0.0 35.83
0.0 0.0 0.0 1.0 35.99
1.0 0.0 0.0 0.0 36.58
0.0 0.0 1.0 0.0 42.92
1.0 0.0 0.0 0.0 37.80
0.0 0.0 1.0 0.0 40.43
0.0 1.0 0.0 0.0 37.89
```

10.3 Program Results

```
 G02DDF Example Program Results

 Residual sum of squares =   0.2223E+02
 Degrees of freedom =    8

 Variable   Parameter estimate   Standard error

      1          0.3600E+02          0.9623E+00
      2          0.3730E+02          0.9623E+00
      3          0.4160E+02          0.9623E+00
      4          0.3788E+02          0.9623E+00
```