NAG Library Routine Document

G02DGF

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

G02DGF calculates the estimates of the parameters of a general linear regression model for a new dependent variable after a call to G02DAF.

2 Specification

```
SUBROUTINE G02DGF (WEIGHT, N, WT, RSS, IP, IRANK, COV, Q, LDQ, SVD, P,     &
                   Y, B, SE, RES, WK, IFAIL)

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

3 Description

G02DGF uses the results given by G02DAF to fit the same set of independent variables to a new dependent variable.

G02DAF computes a QR decomposition of the matrix of p independent variables and also, if the model is not of full rank, a singular value decomposition (SVD). These results can be used to compute estimates of the parameters for a general linear model with a new dependent variable. The QR decomposition leads to the formation of an upper triangular p by p matrix R and an n by n orthogonal matrix Q. In addition the vector $c = Q^{\mathrm{T}}y$ (or $Q^{\mathrm{T}}W^{1/2}y$) is computed. For a new dependent variable, y_{new}, G02DGF computes a new value of $c = Q^{\mathrm{T}}y_{\mathrm{new}}$ or $Q^{\mathrm{T}}W^{1/2}y_{\mathrm{new}}$.

If R is of full rank, then the least squares parameter estimates, $\hat{\beta}$, are 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, then G02DAF will have computed an 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 by G02DAF in the form of the matrix P^*:

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

The matrix Q_* is made available through the workspace of G02DAF.

In addition to parameter estimates, the new residuals are computed and the variance-covariance matrix of the parameter estimates are found by scaling the variance-covariance matrix for the original regression.

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: WEIGHT – CHARACTER(1) *Input*

On entry: indicates if weights are to be used.

WEIGHT = 'U'
Least squares estimation is used.

WEIGHT = 'W'
Weighted least squares is used and weights must be supplied in array WT.

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

2: N – INTEGER *Input*

On entry: n, the number of observations.

Constraint: N $\geq$ IP.

3: WT$(*)$ – REAL (KIND=nag_wp) array *Input*

Note: the dimension of the array WT must be at least N if WEIGHT = 'W', and at least 1 otherwise.

On entry: if WEIGHT = 'W' >, WT must contain the weights to be used in the weighted regression.

If WT$(i) = 0.0$, the ith observation is not included in the model, in which case the effective number of observations is the number of observations with nonzero weights.

If WEIGHT = 'U', WT is not referenced and the effective number of observations is n.

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

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

On entry: the residual sum of squares for the original dependent variable.

On exit: the residual sum of squares for the new dependent variable.

Constraint: RSS > 0.0.

5: IP – INTEGER *Input*

On entry: p, the number of independent variables (including the mean if fitted).

Constraint: $1 \leq$ IP $\leq$ N.

6: IRANK – INTEGER *Input*

On entry: the rank of the independent variables, as given by G02DAF.

Constraint: IRANK > 0, and if SVD = .FALSE., then IRANK = IP, else IRANK $\leq$ IP.

7: COV(IP $\times$ (IP + 1)/2) – REAL (KIND=nag_wp) array *Input/Output*

On entry: the covariance matrix of the parameter estimates as given by G02DAF.

On exit: the upper triangular part of the variance-covariance matrix of the IP 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($(j \times (j-1)/2 + i)$).

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

On entry: the results of the QR decomposition as returned by G02DAF.

On exit: the first column of Q contains the new values of c, the remainder of Q will be unchanged.

9: LDQ – INTEGER *Input*

On entry: the first dimension of the array Q as declared in the (sub)program from which G02DGF is called.

Constraint: LDQ $\geq$ N.

10: SVD – LOGICAL *Input*

On entry: indicates if a singular value decomposition was used by G02DAF.

SVD = .TRUE.
A singular value decomposition was used by G02DAF.

SVD = .FALSE.
A singular value decomposition was not used by G02DAF.

11: P($*$) – REAL (KIND=nag_wp) array *Input*

Note: the dimension of the array P must be at least IP if SVD = .FALSE., and at least IP $\times$ IP + 2 $\times$ IP otherwise.

On entry: details of the QR decomposition and SVD, if used, as returned in array P by G02DAF.

If SVD = .FALSE., only the first IP elements of P are used; these contain the zeta values for the QR decomposition (see F08AEF (DGEQRF) for details).

If SVD = .TRUE., the first IP elements of P contain the zeta values for the QR decomposition (see F08AEF (DGEQRF) for details) and the next IP $\times$ IP + IP elements of P contain details of the singular value decomposition.

12: Y(N) – REAL (KIND=nag_wp) array *Input*

On entry: the new dependent variable, y_{new}.

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

On exit: the least squares estimates of the parameters of the regression model, $\hat{\beta}$.

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

On exit: the standard error of the estimates of the parameters.

15: RES(N) – REAL (KIND=nag_wp) array *Output*

On exit: the residuals for the new regression model.

16: WK($5 \times$ (IP $-$ 1) + IP $\times$ IP) – REAL (KIND=nag_wp) array *Input*

On entry: if SVD = .TRUE., WK must be unaltered from the previous call to G02DAF or G02DGF.

If SVD = .FALSE., WK is used as workspace.

17: 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, IP $<$ 1,
or N $<$ IP,
or IRANK $\leq$ 0,
or SVD $=$ .FALSE. and IRANK $\neq$ IP,
or SVD $=$ .TRUE. and IRANK $>$ IP,
or LDQ $<$ N,
or RSS $\leq$ 0.0,
or WEIGHT $\neq$ 'U' or 'W'.

IFAIL $= 2$

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

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 same accuracy as G02DAF is obtained.

8 Parallelism and Performance

G02DGF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

G02DGF 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

The values of the leverages, h_i, are unaltered by a change in the dependent variable so a call to G02FAF can be made using the value of H from G02DAF.

10 Example

A dataset consisting of 12 observations with four independent variables and two dependent variables are read in. A model with all four independent variables is fitted to the first dependent variable by G02DAF and the results printed. The model is then fitted to the second dependent variable by G02DGF and those results printed.

10.1 Program Text

```
    Program g02dgfe

!     G02DGF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: g02daf, g02dgf, 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, ldx,  &
                                          lwk, lwt, m, n
      Logical                          :: svd
      Character (1)                    :: mean, weight
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: b(:), cov(:), h(:), oy(:), p(:),     &
                                          q(:,:), res(:), se(:), wk(:), wt(:), &
                                          x(:,:), y(:)
      Integer, Allocatable             :: isx(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: count
!     .. Executable Statements ..
      Write (nout,*) 'G02DGF Example Program Results'
      Write (nout,*)

!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n, m, weight, mean

      If (weight=='W' .Or. weight=='w') Then
        lwt = n
      Else
        lwt = 0
      End If
      ldx = n
      Allocate (x(ldx,m),isx(m),oy(n),y(n),wt(lwt))

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

```
      Else
        Read (nin,*)(x(i,1:m),oy(i),i=1,n)
      End If

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

!     Calculate IP
      ip = count(isx(1:m)>0)
      If (mean=='M' .Or. mean=='m') Then
        ip = ip + 1
      End If

      lwk = 5*(ip-1) + ip*ip
      ldq = n
      Allocate (b(ip),se(ip),cov(ip*(ip+1)/2),res(n),h(n),q(ldq,ip+1),p(2*ip+ &
        ip*ip),wk(lwk))

!     Use suggested value for tolerance
      tol = 0.000001E0_nag_wp

!     Fit general linear regression model to first dependent variable
      ifail = 0
      Call g02daf(mean,weight,n,x,ldx,m,isx,ip,oy,wt,rss,idf,b,se,cov,res,h,q, &
        ldq,svd,irank,p,tol,wk,ifail)

!     Display results for model fit to original dependent variable
      Write (nout,*) 'Results for original y-variable using G02DAF'
      Write (nout,*)
      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)
      Write (nout,*)

!     Read in the new dependent variable
      Read (nin,*) y(1:n)

!     Fit same model to different dependent variable
      ifail = 0
      Call g02dgf(weight,n,wt,rss,ip,irank,cov,q,ldq,svd,p,y,b,se,res,wk, &
        ifail)

!     Display results for model fit to new dependent variable
      Write (nout,*) 'Results for second y-variable using G02DGF'
      Write (nout,*)
      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)

99999 Format (1X,A,E12.4)
99998 Format (1X,A,I4)
99997 Format (1X,I6,2E20.4)
    End Program g02dgfe
```

10.2 Program Data

```
G02DGF Example Program Data
 12 4 'U' 'M'                  :: N, M, MEAN, WEIGHT
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          :: End of X, OY (original dependent variable)
 1   1   1   1                 :: ISX
63.0 69.0 68.0 71.0 68.0 65.0
65.0 66.0 72.0 67.0 70.0 67.0  :: Y (new dependent variable)
```

10.3 Program Results

```
 G02DGF Example Program Results

 Results for original y-variable using G02DAF

 Model not of full rank

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

 Variable   Parameter estimate   Standard error

      1          0.3056E+02          0.3849E+00
      2          0.5447E+01          0.8390E+00
      3          0.6743E+01          0.8390E+00
      4          0.1105E+02          0.8390E+00
      5          0.7320E+01          0.8390E+00

 Results for second y-variable using G02DGF

 Residual sum of squares =   0.2400E+02
 Degrees of freedom =    8

 Variable   Parameter estimate   Standard error

      1          0.5407E+02          0.4000E+00
      2          0.1127E+02          0.8718E+00
      3          0.1260E+02          0.8718E+00
      4          0.1693E+02          0.8718E+00
      5          0.1327E+02          0.8718E+00
```