

NAG Library Routine Document

G02DEF

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

G02DEF adds a new independent variable to a general linear regression model.

2 Specification

```
SUBROUTINE G02DEF (WEIGHT, N, IP, Q, LDQ, P, WT, X, RSS, TOL, IFAIL)
  INTEGER          N, IP, LDQ, IFAIL
  REAL (KIND=nag_wp) Q(LDQ,IP+2), P(IP+1), WT(*), X(N), RSS, TOL
  CHARACTER(1)     WEIGHT
```

3 Description

A linear regression model may be built up by adding new independent variables to an existing model. G02DEF updates the QR decomposition used in the computation of the linear regression model. The QR decomposition may come from G02DAF or a previous call to G02DEF. The general linear regression model is defined by

$$y = X\beta + \epsilon,$$

where y is a vector of n observations on the dependent variable,

X is an n by p matrix of the independent variables of column rank k ,

β is a vector of length p of unknown arguments,

and ϵ is a vector of length n of unknown random errors such that $\text{var } \epsilon = V\sigma^2$, where V is a known diagonal matrix.

If $V = I$, the identity matrix, then least squares estimation is used. If $V \neq I$, then for a given weight matrix $W \propto V^{-1}$, weighted least squares estimation is used.

The least squares estimates, $\hat{\beta}$ of the arguments β minimize $(y - X\beta)^T(y - X\beta)$ while the weighted least squares estimates, minimize $(y - X\beta)^T W(y - X\beta)$.

The parameter estimates may be found by computing a QR decomposition of X (or $W^{\frac{1}{2}}X$ in the weighted case), i.e.,

$$X = QR^* \quad \left(\text{or} \quad W^{\frac{1}{2}}X = QR^* \right),$$

where $R^* = \begin{pmatrix} R \\ 0 \end{pmatrix}$ and R is a p by p upper triangular matrix and Q is an n by n orthogonal matrix.

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

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

where $c = Q^T y$ (or $Q^T W^{\frac{1}{2}} y$) and 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 .

To add a new independent variable, x_{p+1} , R and c have to be updated. The matrix Q_{p+1} is found such that $Q_{p+1}^T [R : Q^T x_{p+1}]$ (or $Q_{p+1}^T [R : Q^T W^{\frac{1}{2}} x_{p+1}]$) is upper triangular. The vector c is then updated by multiplying by Q_{p+1}^T .

The new independent variable is tested to see if it is linearly related to the existing independent variables by checking that at least one of the values $(Q^T x_{p+1})_i$, for $i = p + 2, \dots, n$, is nonzero.

The new parameter estimates, $\hat{\beta}$, can then be obtained by a call to G02DDF.

The routine can be used with $p = 0$, in which case R and c are initialized.

4 References

Draper N R and Smith H (1985) *Applied Regression Analysis* (2nd Edition) Wiley

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

McCullagh P and Nelder J A (1983) *Generalized Linear Models* Chapman and Hall

Searle S R (1971) *Linear Models* Wiley

5 Arguments

- 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: $N \geq 1$.
- 3: IP – INTEGER *Input*
On entry: p , the number of independent variables already in the model.
Constraint: $IP \geq 0$ and $IP < N$.
- 4: Q(LDQ, IP + 2) – REAL (KIND=nag_wp) array *Input/Output*
On entry: if $IP \neq 0$, Q must contain the results of the QR decomposition for the model with p arguments as returned by G02DAF or a previous call to G02DEF.
 If $IP = 0$, the first column of Q should contain the n values of the dependent variable, y .
On exit: the results of the QR decomposition for the model with $p + 1$ arguments:
 the first column of Q contains the updated value of c ;
 the columns 2 to IP + 1 are unchanged;
 the first IP + 1 elements of column IP + 2 contain the new column of R , while the remaining $N - IP - 1$ elements contain details of the matrix Q_{p+1} .

- 5: LDQ – INTEGER Input
On entry: the first dimension of the array Q as declared in the (sub)program from which G02DEF is called.
Constraint: $LDQ \geq N$.
- 6: P(IP + 1) – REAL (KIND=nag_wp) array Input/Output
On entry: contains further details of the *QR* decomposition used. The first IP elements of P must contain the zeta values for the *QR* decomposition (see F08AEF (DGEQRF) for details).
The first IP elements of array P are provided by G02DAF or by previous calls to G02DEF.
On exit: the first IP elements of P are unchanged and the (IP + 1)th element contains the zeta value for Q_{p+1} .
- 7: 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.
If $WT(i) = 0.0$, the *i*th 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 $i = 1, 2, \dots, n$.
- 8: X(N) – REAL (KIND=nag_wp) array Input
On entry: *x*, the new independent variable.
- 9: RSS – REAL (KIND=nag_wp) Output
On exit: the residual sum of squares for the new fitted model.
Note: this will only be valid if the model is of full rank, see Section 9.
- 10: TOL – REAL (KIND=nag_wp) Input
On entry: the value of TOL is used to decide if the new independent variable is linearly related to independent variables already included in the model. If the new variable is linearly related then *c* is not updated. The smaller the value of TOL the stricter the criterion for deciding if there is a linear relationship.
Suggested value: $TOL = 0.000001$.
Constraint: $TOL > 0.0$.
- 11: 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, because for this routine the values of the output arguments may be useful even if $IFAIL \neq 0$ on exit, the recommended value is -1. **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).

Note: G02DEF may return useful information for one or more of the following detected errors or warnings.

Errors or warnings detected by the routine:

$IFAIL = 1$

On entry, $N < 1$,
 or $IP < 0$,
 or $IP \geq N$,
 or $LDQ < N$,
 or $TOL \leq 0.0$,
 or $WEIGHT \neq 'U'$ or $'W'$.

$IFAIL = 2$

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

$IFAIL = 3$

The new independent variable is a linear combination of existing variables. The $(IP + 2)$ th column of Q will therefore be null.

$IFAIL = -99$

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 is closely related to the accuracy of F08AGF (DORMQR) which should be consulted for further details.

8 Parallelism and Performance

G02DEF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

G02DEF 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

It should be noted that the residual sum of squares produced by G02DEF may not be correct if the model to which the new independent variable is added is not of full rank. In such a case G02DDF should be used to calculate the residual sum of squares.

10 Example

A dataset consisting of 12 observations is read in. The four independent variables are stored in the array X while the dependent variable is read into the first column of Q. If the character variable *mean* indicates that a mean should be included in the model a variable taking the value 1.0 for all observations is set up and fitted. Subsequently, one variable at a time is selected to enter the model as indicated by the input value of *indx*. After the variable has been added the parameter estimates are calculated by G02DDF and the results printed. This is repeated until the input value of *indx* is 0.

10.1 Program Text

```

Program g02defe

!      G02DEF Example Program Text

!      Mark 26 Release. NAG Copyright 2016.

!      .. 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, rsst, 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,*) 'G02DEF 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))

!      Read in the dependent variable, Y, and store in first column of Q
      Read (nin,*) q(1:n,1)

!      Read in weights
      If (lwt>0) Then
         Read (nin,*) wt(1:n)
      End If

!      Use suggested value for tolerance
      tol = 0.000001E0_nag_wp

```

```

!      Loop over each of the supplied variables
      ip = 0
u_lp: Do
      Read (nin,*,Iostat=ifail) x(1:n)
      If (ifail/=0) Then
        Exit u_lp
      End If

!      Add the new variable to the model
      ifail = -1
      Call g02def(weight,n,ip,q,ldq,p,wt,x,rss,tol,ifail)
      If (ifail/=0) Then
        If (ifail==3) Then
          Write (nout,99999) ' * Variable ', ip, &
            ' is linear combination of previous columns'
          Write (nout,*) ' so it has not been added'
          Write (nout,*)
          Cycle u_lp
        Else
          Go To 100
        End If
      End If

      ip = ip + 1
      Write (nout,99999) 'Variable ', ip, ' added'

!      Get G02DDF to recalculate RSS
      rsst = 0.0E0_nag_wp

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

      If (svd) Then
        Write (nout,*) 'Model not of full rank'
        Write (nout,*)
      End If
      Write (nout,99998) 'Residual sum of squares = ', rsst
      Write (nout,99999) '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,*)
End Do u_lp

100  Continue

99999 Format (1X,A,I0,A)
99998 Format (1X,A,E13.4)
99997 Format (1X,I6,2E20.4)
      End Program g02defe

```

10.2 Program Data

G02DEF Example Program Data

```

12 5 'U' :: N, M (max. number of variables), WEIGHT
4.32 5.21 6.49 7.10 7.94 8.53
8.84 9.02 9.27 9.43 9.68 9.83 :: End of Y
1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 :: End of X0 (intercept)
1.0 1.5 2.0 2.5 3.0 3.5
4.0 4.5 5.0 5.5 6.0 6.5 :: End of X1
0.0 0.0 0.0 0.0 0.0 0.0
1.0 1.0 1.0 1.0 1.0 1.0 :: End of X2
0.0 0.0 0.0 0.0 0.0 0.0
4.0 4.5 5.0 5.5 6.0 6.5 :: End of X3
1.4 2.2 4.5 6.1 7.1 7.7
8.3 8.6 8.8 9.0 9.3 9.2 :: End of X4

```

10.3 Program Results

G02DEF Example Program Results

Variable 1 added

Residual sum of squares = 0.3627E+02

Degrees of freedom = 11

Variable	Parameter estimate	Standard error
1	0.7972E+01	0.5242E+00

Variable 2 added

Residual sum of squares = 0.4016E+01

Degrees of freedom = 10

Variable	Parameter estimate	Standard error
1	0.4410E+01	0.4376E+00
2	0.9498E+00	0.1060E+00

Variable 3 added

Residual sum of squares = 0.3887E+01

Degrees of freedom = 9

Variable	Parameter estimate	Standard error
1	0.4224E+01	0.5673E+00
2	0.1055E+01	0.2222E+00
3	-0.4196E+00	0.7670E+00

Variable 4 added

Residual sum of squares = 0.1870E+00

Degrees of freedom = 8

Variable	Parameter estimate	Standard error
1	0.2760E+01	0.1759E+00
2	0.1706E+01	0.7310E-01
3	0.4458E+01	0.4268E+00
4	-0.1301E+01	0.1034E+00

Variable 5 added

Residual sum of squares = 0.8407E-01

Degrees of freedom = 7

Variable	Parameter estimate	Standard error
1	0.3144E+01	0.1818E+00
2	0.9075E+00	0.2776E+00
3	0.2079E+01	0.8680E+00
4	-0.6159E+00	0.2453E+00
5	0.2922E+00	0.9981E-01
