

# NAG Library Routine Document

## G02DNF

**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

G02DNF gives the estimate of an estimable function along with its standard error.

### 2 Specification

```

SUBROUTINE G02DNF (IP, IRANK, B, COV, P, F, EST, STAT, SESTAT, T, TOL,      &
                  WK, IFAIL)
INTEGER            IP, IRANK, IFAIL
REAL (KIND=nag_wp) B(IP), COV(IP*(IP+1)/2), P(IP*IP+2*IP), F(IP),      &
                  STAT, SESTAT, T, TOL, WK(IP)
LOGICAL           EST

```

### 3 Description

G02DNF computes the estimates of an estimable function for a general linear regression model which is not of full rank. It is intended for use after a call to G02DAF or G02DDF. An estimable function is a linear combination of the arguments such that it has a unique estimate. For a full rank model all linear combinations of arguments are estimable.

In the case of a model not of full rank the routines use a singular value decomposition (SVD) to find the parameter estimates,  $\hat{\beta}$ , and their variance-covariance matrix. Given the upper triangular matrix  $R$  obtained from the  $QR$  decomposition of the independent variables the SVD gives

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

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

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

$P_1$  being the first  $k$  columns of  $P$ , i.e.,  $P = (P_1 P_0)$ ,  $Q_{*1}$  being the first  $k$  columns of  $Q_*$ , and  $c_1$  being the first  $p$  elements of  $c$ .

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

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

as given by G02DAF and G02DDF.

A linear function of the arguments,  $F = f^T \beta$ , can be tested to see if it is estimable by computing  $\zeta = P_0^T f$ . If  $\zeta$  is zero, then the function is estimable; if not, the function is not estimable. In practice  $|\zeta|$  is tested against some small quantity  $\eta$ .

Given that  $F$  is estimable it can be estimated by  $f^T \hat{\beta}$  and its standard error calculated from the variance-covariance matrix of  $\hat{\beta}$ ,  $C_\beta$ , as

$$\text{se}(F) = \sqrt{f^T C_\beta f}.$$

Also a  $t$ -statistic,

$$t = \frac{f^T \hat{\beta}}{\text{se}(F)},$$

can be computed. The  $t$ -statistic will have a Student's  $t$ -distribution with degrees of freedom as given by the degrees of freedom for the residual sum of squares for the model.

## 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 Arguments

- 1: IP – INTEGER *Input*  
*On entry:*  $p$ , the number of terms in the linear model.  
*Constraint:*  $IP \geq 1$ .
  
- 2: IRANK – INTEGER *Input*  
*On entry:*  $k$ , the rank of the independent variables.  
*Constraint:*  $1 \leq IRANK \leq IP$ .
  
- 3: B(IP) – REAL (KIND=nag\_wp) array *Input*  
*On entry:* the IP values of the estimates of the arguments of the model,  $\hat{\beta}$ .
  
- 4: COV(IP × (IP + 1)/2) – REAL (KIND=nag\_wp) array *Input*  
*On entry:* 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( $(j \times (j - 1)/2 + i)$ ).
  
- 5: P(IP × IP + 2 × IP) – REAL (KIND=nag\_wp) array *Input*  
*On entry:* as returned by G02DAF and G02DDF.
  
- 6: F(IP) – REAL (KIND=nag\_wp) array *Input*  
*On entry:*  $f$ , the linear function to be estimated.
  
- 7: EST – LOGICAL *Output*  
*On exit:* indicates if the function was estimable.  
EST = .TRUE.  
The function is estimable.  
EST = .FALSE.  
The function is not estimable and STAT, SESTAT and T are not set.
  
- 8: STAT – REAL (KIND=nag\_wp) *Output*  
*On exit:* if EST = .TRUE., STAT contains the estimate of the function,  $f^T \hat{\beta}$ .

- 9: SESTAT – REAL (KIND=nag\_wp) Output  
*On exit:* if EST = .TRUE., SESTAT contains the standard error of the estimate of the function,  $se(F)$ .
- 10: T – REAL (KIND=nag\_wp) Output  
*On exit:* if EST = .TRUE., T contains the  $t$ -statistic for the test of the function being equal to zero.
- 11: TOL – REAL (KIND=nag\_wp) Input  
*On entry:*  $\eta$ , the tolerance value used in the check for estimability.  
 $TOL \leq 0.0$   
 $\sqrt{\epsilon}$ , where  $\epsilon$  is the *machine precision*, is used instead.
- 12: WK(IP) – REAL (KIND=nag\_wp) array Workspace
- 13: 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:** G02DNF 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, IP < 1,  
 or IRANK < 1,  
 or IRANK > IP.

IFAIL = 2

On entry, IRANK = IP. In this case EST is returned as true and all statistics are calculated.

IFAIL = 3

Standard error of statistic = 0.0; this may be due to rounding errors if the standard error is very small or due to mis-specified inputs COV and F.

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 computations are believed to be stable.

## 8 Parallelism and Performance

G02DNF 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 value of estimable functions is independent of the solution chosen from the many possible solutions. While G02DNF may be used to estimate functions of the arguments of the model as computed by G02DKF,  $\beta_c$ , these must be expressed in terms of the original arguments,  $\beta$ . The relation between the two sets of arguments may not be straightforward.

## 10 Example

Data from an experiment with four treatments and three observations per treatment are read in. A model, with a mean term, is fitted by G02DAF. The number of functions to be tested is read in, then the linear functions themselves are read in and tested with G02DNF. The results of G02DNF are printed.

### 10.1 Program Text

```

Program g02dnfe
!      G02DNF Example Program Text
!      Mark 26 Release. NAG Copyright 2016.
!
!      .. Use Statements ..
      Use nag_library, Only: g02daf, g02dnf, nag_wp
!      .. Implicit None Statement ..
      Implicit None
!      .. Parameters ..
      Integer, Parameter          :: nin = 5, nout = 6
!      .. Local Scalars ..
      Real (Kind=nag_wp)          :: rss, sestat, stat, t, tol
      Integer                     :: i, idf, ifail, ip, irank, ldq, ldx, &
                                   lwk, lwt, m, n
      Logical                     :: est, svd
      Character (1)               :: mean, weight
!      .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable :: b(:), cov(:), f(:), h(:), p(:),      &
                                   q(:, :), res(:), se(:), wk(:), wt(:), &
                                   x(:, :), y(:)
      Integer, Allocatable        :: isx(:)

```

```

!      .. Intrinsic Procedures ..
      Intrinsic                                :: count, max
!      .. Executable Statements ..
      Write (nout,*) 'G02DNF Example Program Results'
      Write (nout,*)

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

!      Read in the problem size
      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),y(n),wt(lwt))

      If (lwt>0) Then
         Read (nin,*)(x(i,1:m),y(i),wt(i),i=1,n)
      Else
         Read (nin,*)(x(i,1:m),y(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 = max(5*(ip-1)+ip*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),f(ip))

!      Use suggested value for tolerance
      tol = 0.000001E0_nag_wp

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

!      Display initial parameter estimates
      Write (nout,*) 'Estimates from G02DAF'
      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)
      Write (nout,*)

      i = 0
estfn_lp: Do
      Read (nin,*,Iostat=ifail) f(1:ip)
      If (ifail/=0) Then
         Exit estfn_lp
      End If

      i = i + 1

!      Compute the estimable function
      ifail = -1
      Call g02dnf(ip,irank,b,cov,p,f,est,stat,sestat,t,tol,wk,ifail)

```

```

      If (ifail/=0) Then
        If (ifail/=2) Then
          Go To 100
        End If
      End If

!      Display results
      Write (nout,99996) 'Function ', i
      Write (nout,*)
      Write (nout,99995) f(1:ip)
      Write (nout,*)
      If (est) Then
        Write (nout,99994) 'STAT = ', stat, ' SE = ', sestat, ' T = ', t
      Else
        Write (nout,*) 'Function not estimable'
      End If
      Write (nout,*)
    End Do estfn_lp

100   Continue

99999 Format (1X,A,E12.4)
99998 Format (1X,A,I4)
99997 Format (1X,I6,2E20.4)
99996 Format (1X,A,I4)
99995 Format (1X,5F8.2)
99994 Format (1X,A,F10.4,A,F10.4,A,F10.4)
      End Program g02dnfe

```

## 10.2 Program Data

```

G02DNF Example Program Data
 12 4 'U' 'M'           :: N, M, WEIGHT, MEAN
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, Y
 1 1 1 1               :: ISX
 1.0 1.0 0.0 0.0 0.0   :: Estimable function 1
 0.0 1.0 -1.0 0.0 0.0  :: Estimable function 2
 0.0 1.0 0.0 0.0 0.0  :: Estimable function 3

```

## 10.3 Program Results

G02DNF Example Program Results

Estimates from G02DAF

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

Function	1
1.00	1.00 0.00 0.00 0.00

STAT = 36.0033 SE = 0.9623 T = 37.4119

Function 2

0.00 1.00 -1.00 0.00 0.00

STAT = -1.2967 SE = 1.3610 T = -0.9528

Function 3

0.00 1.00 0.00 0.00 0.00

Function not estimable

---