

NAG Library Routine Document

G02GNF

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

G02GNF gives the estimate of an estimable function along with its standard error from the results from fitting a generalized linear model.

2 Specification

```

SUBROUTINE G02GNF (IP, IRANK, B, COV, V, LDV, F, EST, STAT, SESTAT, Z,      &
                  TOL, WK, IFAIL)
INTEGER            IP, IRANK, LDV, IFAIL
REAL (KIND=nag_wp) B(IP), COV(IP*(IP+1)/2), V(LDV,IP+7), F(IP), STAT,      &
                  SESTAT, Z, TOL, WK(IP)
LOGICAL           EST

```

3 Description

G02GNF computes the estimates of an estimable function for a generalized linear model which is not of full rank. It is intended for use after a call to G02GAF, G02GBF, G02GCF or G02GDF. An estimable function is a linear combination of the parameters such that it has a unique estimate. For a full rank model all linear combinations of parameters 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 leads to a 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 described by G02GAF, G02GBF, G02GCF and G02GDF.

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

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_β , as

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

Also a z statistic

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

can be computed. The distribution of z will be approximately Normal.

4 References

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

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

Searle S R (1971) *Linear Models* Wiley

5 Parameters

- 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 dependent 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 parameters 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: V(LDV, IP + 7) – REAL (KIND=nag_wp) array *Input*
On entry: as returned by G02GAF, G02GBF, G02GCF and G02GDF.

- 6: LDV – INTEGER *Input*
On entry: the first dimension of the array V as declared in the (sub)program from which G02GNF is called.
Constraint: $LDV \geq IP$.

- 7: F(IP) – REAL (KIND=nag_wp) array *Input*
On entry: f , the linear function to be estimated.

- 8: 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 Z are not set.

- 9: STAT – REAL (KIND=nag_wp) Output
On exit: if EST = .TRUE., STAT contains the estimate of the function, $f^T \hat{\beta}$
- 10: SESTAT – REAL (KIND=nag_wp) Output
On exit: if EST = .TRUE., SESTAT contains the standard error of the estimate of the function, $se(F)$.
- 11: Z – REAL (KIND=nag_wp) Output
On exit: if EST = .TRUE., Z contains the z statistic for the test of the function being equal to zero.
- 12: TOL – REAL (KIND=nag_wp) Input
On entry: the tolerance value used in the check for estimability, η .
 If $TOL \leq 0.0$ then $\sqrt{\epsilon}$, where ϵ is the *machine precision*, is used instead.
- 13: WK(IP) – REAL (KIND=nag_wp) array Workspace
- 14: 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, because for this routine the values of the output parameters 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: G02GNF 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,
 or LDV < 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 3.8 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 computations are believed to be stable.

8 Parallelism and Performance

G02GNF is not threaded by NAG in any implementation.

G02GNF 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 G02GNF may be used to estimate functions of the parameters of the model as computed by G02GKF, β_c , these must be expressed in terms of the original parameters, β . The relation between the two sets of parameters may not be straightforward.

10 Example

A loglinear model is fitted to a 3 by 5 contingency table by G02GCF. The model consists of terms for rows and columns. The table is:

141	67	114	79	39
131	66	143	72	35
36	14	38	28	16

The number of functions to be tested is read in, then the linear functions themselves are read in and tested with G02GNF. The results of G02GNF are printed.

10.1 Program Text

```

Program g02gnfe
!      G02GNF Example Program Text
!
!      Mark 25 Release. NAG Copyright 2014.
!
!      .. Use Statements ..
!      Use nag_library, Only: g02gcf, g02gnf, nag_wp
!      .. Implicit None Statement ..

```

```

Implicit None
! .. Parameters ..
Integer, Parameter          :: nin = 5, nout = 6
! .. Local Scalars ..
Real (Kind=nag_wp)         :: a, dev, eps, sestat, stat, tol, z
Integer                    :: i, idf, ifail, ip, iprint, irank,    &
                          ldv, ldx, lwk, lwt, m, maxit, n
Logical                    :: est
Character (1)              :: link, mean, offset, weight
! .. Local Arrays ..
Real (Kind=nag_wp), Allocatable :: b(:), cov(:), f(:), se(:), v(:, :),    &
                          wk(:), wt(:), x(:, :), y(:)
Integer, Allocatable       :: isx(:)
! .. Intrinsic Procedures ..
Intrinsic                  :: count, max
! .. Executable Statements ..
Write (nout,*) 'G02GNF Example Program Results'
Write (nout,*)

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

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

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

! Read in data
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

! Read in power for exponential link
If (link=='E' .Or. link=='e') Then
  Read (nin,*) a
End If

ldv = n
lwk = max((ip*ip+3*ip+22)/2,ip)
Allocate (b(ip),se(ip),cov(ip*(ip+1)/2),v(ldv,ip+7),wk(lwk),f(ip))

! Read in the offset
If (offset=='Y' .Or. offset=='y') Then
  Read (nin,*) v(1:n,7)
End If

! Read in control parameters
Read (nin,*) iprint, eps, tol, maxit

! Fit generalized linear model with Poisson errors
ifail = -1
Call g02gcf('L','M','N','U',n,x,ldx,m,isx,ip,y,wt,a,dev,idf,b,irank,se, &
          cov,v,ldv,tol,maxit,iprint,eps,wk,ifail)
If (ifail/=0) Then

```

```

      If (ifail<7) Then
        Go To 100
      End If
    End If

!   Display initial results
    Write (nout,99999) 'Deviance = ', dev
    Write (nout,99998) 'Degrees of freedom = ', idf
    Write (nout,*)
    Write (nout,*) '      Estimate      Standard error'
    Write (nout,*)
    Write (nout,99997)(b(i),se(i),i=1,ip)

!   Estimate the estimable functions
    i = 0
fun_lp: Do
!   Read in the function
    Read (nin,*,Iostat=ifail) f(1:ip)
    If (ifail/=0) Then
      Exit fun_lp
    End If

    i = i + 1

!   Estimate it
    ifail = -1
    Call g02gnf(ip,irank,b,cov,v,ldv,f,est,stat,sestat,z,tol,wk,ifail)
    If (ifail/=0) Then
      If (ifail/=2) Then
        Go To 100
      End If
    End If

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

100  Continue

99999 Format (1X,A,E12.4)
99998 Format (1X,A,I2)
99997 Format (1X,2F14.4)
99996 Format (1X,A,I4)
99995 Format (1X,5F8.2)
99994 Format (1X,A,F10.4,A,F10.4,A,F10.4)
      End Program g02gnfe

```

10.2 Program Data

G02GNF Example Program Data

```

'L' 'M' 'N' 'U' 15 8      :: LINK,MEAN,OFFSET,WEIGHT,N,M
1.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 141.0
1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 67.0
1.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 114.0
1.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 79.0
1.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 39.0
0.0 1.0 0.0 1.0 0.0 0.0 0.0 0.0 131.0
0.0 1.0 0.0 0.0 1.0 0.0 0.0 0.0 66.0
0.0 1.0 0.0 0.0 0.0 1.0 0.0 0.0 143.0
0.0 1.0 0.0 0.0 0.0 0.0 1.0 0.0 72.0
0.0 1.0 0.0 0.0 0.0 0.0 0.0 1.0 35.0
0.0 0.0 1.0 1.0 0.0 0.0 0.0 0.0 36.0

```

```

0.0 0.0 1.0 0.0 1.0 0.0 0.0 0.0 14.0
0.0 0.0 1.0 0.0 0.0 1.0 0.0 0.0 38.0
0.0 0.0 1.0 0.0 0.0 0.0 1.0 0.0 28.0
0.0 0.0 1.0 0.0 0.0 0.0 0.0 1.0 16.0  :: End of X, Y
  1  1  1  1  1  1  1  1  :: ISX
  0 1.0E-6 5.0E-5 0  :: IPRINT,EPS,TOL,MAXIT
  1.0 1.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0  :: Estimable function 1 (F)
  0.0 1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0  :: Estimable function 2 (F)
  0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  :: Estimable function 3 (F)

```

10.3 Program Results

G02GNF Example Program Results

Deviance = 0.9038E+01
Degrees of freedom = 8

Estimate	Standard error
2.5977	0.0258
1.2619	0.0438
1.2777	0.0436
0.0580	0.0668
1.0307	0.0551
0.2910	0.0732
0.9876	0.0559
0.4880	0.0675
-0.1996	0.0904

Function	1				
1.00	1.00	0.00	0.00	1.00	
0.00	0.00	0.00	0.00		

STAT = 4.8903 SE = 0.0674 z = 72.5934

Function	2				
0.00	1.00	-1.00	0.00	0.00	
0.00	0.00	0.00	0.00		

STAT = -0.0158 SE = 0.0672 z = -0.2350

Function	3				
0.00	1.00	0.00	0.00	0.00	
0.00	0.00	0.00	0.00		

Function not estimable
