

## G12BAF – NAG Fortran Library Routine Document

**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

G12BAF returns parameter estimates and other statistics that are associated with the Cox proportional hazards model for fixed covariates.

### 2 Specification

```

SUBROUTINE G12BAF(OFFSET, N, M, NS, Z, LDZ, ISZ, IP, T, IC, OMEGA,
1             ISI, DEV, B, SE, SC, COV, RES, ND, TP, SUR,
2             NDMAX, TOL, MAXIT, IPRINT, WK, IWK, IFAIL)
  INTEGER
    N, M, NS, LDZ, ISZ(M), IP, IC(N), ISI(*), ND,
1     NDMAX, MAXIT, IPRINT, IWK(2*N), IFAIL
  real
    Z(LDZ,M), T(N), OMEGA(*), DEV, B(IP), SE(IP),
1     SC(IP), COV(IP*(IP+1)/2), RES(N), TP(NDMAX),
2     SUR(NDMAX,*), TOL, WK(IP*(IP+9)/2+N)
  CHARACTER*1  OFFSET

```

### 3 Description

The proportional hazard model relates the time to an event, usually death or failure, to a number of explanatory variables known as covariates. Some of the observations may be right censored, that is the exact time to failure is not known, only that it is greater than a known time.

Let  $t_i, i = 1, \dots, n$  be the the failure time or censored time for the  $i$ th observation with the vector of  $p$  covariates  $z_i$ . It is assumed that censoring and failure mechanisms are independent. The hazard function,  $\lambda(t, z)$ , is the probability that an individual with covariates  $z$  fails at time  $t$  given that the individual survived up to time  $t$ . In the Cox proportional hazards model (Cox [1])  $\lambda(t, z)$  is of the form:

$$\lambda(t, z) = \lambda_0(t) \exp(z^T \beta + \omega)$$

where  $\lambda_0$  is the base-line hazard function, an unspecified function of time,  $\beta$  is a vector of unknown parameters and  $\omega$  is a known offset.

Assuming there are ties in the failure times giving  $n_d < n$  distinct failure times,  $t_{(1)} < \dots < t_{(n_d)}$  such that  $d_i$  individuals fail at  $t_{(i)}$ , it follows that the marginal likelihood for  $\beta$  is well approximated (see Kalbfleisch and Prentice, [2]) by:

$$L = \prod_{i=1}^{n_d} \frac{\exp(s_i^T \beta + \omega_i)}{[\sum_{l \in R(t_{(i)})} \exp(z_l^T \beta + \omega_l)]^{d_i}} \quad (1)$$

where  $s_i$  is the sum of the covariates of individuals observed to fail at  $t_{(i)}$  and  $R(t_{(i)})$  is the set of individuals at risk just prior to  $t_{(i)}$ , that is it is all individuals that fail or are censored at time  $t_{(i)}$  along with all individuals that survive beyond time  $t_{(i)}$ . The maximum likelihood estimates (MLEs) of  $\beta$ , given by  $\hat{\beta}$ , are obtained by maximizing (1) using a Newton–Raphson iteration technique that includes step halving and utilizes the first and second partial derivatives of (1) which are given by equations (2) and (3) below:

$$U_j(\beta) = \frac{\partial \ln L}{\partial \beta_j} = \sum_{i=1}^{n_d} [s_{ji} - d_i \alpha_{ji}(\beta)] = 0 \quad (2)$$

for  $j = 1, \dots, p$ , where  $s_{ji}$  is the  $j$ th element in the vector  $s_i$  and

$$\alpha_{ji}(\beta) = \frac{\sum_{l \in R(t_{(i)})} z_{jl} \exp(z_l^T \beta + \omega_l)}{\sum_{l \in R(t_{(i)})} \exp(z_l^T \beta + \omega_l)}.$$

Similarly,

$$I_{hj}(\beta) = -\frac{\partial^2 \ln L}{\partial \beta_h \partial \beta_j} = \sum_{i=1}^{n_d} d_i \gamma_{hji} \quad (3)$$

where

$$\gamma_{hji} = \frac{\sum_{l \in R(t_{(i)})} z_{hl} z_{jl} \exp(z_l^T \beta + \omega_l)}{\sum_{l \in R(t_{(i)})} \exp(z_l^T \beta + \omega_l)} - \alpha_{hi}(\beta) \alpha_{ji}(\beta) \quad h, j = 1, \dots, p.$$

$U_j(\beta)$  is the  $j$ th component of a score vector and  $I_{hj}(\beta)$  is the  $(h, j)$  element of the observed information matrix  $I(\beta)$  whose inverse  $I(\beta)^{-1} = [I_{hj}(\beta)]^{-1}$  gives the variance-covariance matrix of  $\beta$ .

It should be noted that if a covariate or a linear combination of covariates is monotonically increasing or decreasing with time then one or more of the  $\beta_j$ 's will be infinite.

If  $\lambda_0(t)$  varies across  $\nu$  strata, where the number of individuals in the  $k$ th stratum is  $n_k$   $k = 1, \dots, \nu$  with  $n = \sum_{k=1}^{\nu} n_k$ , then rather than maximizing (1) to obtain  $\hat{\beta}$ , the following marginal likelihood is maximized:

$$L = \prod_{k=1}^{\nu} L_k, \quad (4)$$

where  $L_k$  is the contribution to likelihood for the  $n_k$  observations in the  $k$ th stratum treated as a single sample in (1). When strata are included the covariate coefficients are constant across strata but there is a different base-line hazard function  $\lambda_0$ .

The base-line survivor function associated with a failure time  $t_{(i)}$ , is estimated as  $\exp(-\hat{H}(t_{(i)}))$ , where

$$\hat{H}(t_{(i)}) = \sum_{t_{(j)} \leq t_{(i)}} \left( \frac{d_i}{\sum_{l \in R(t_{(j)})} \exp(z_l^T \hat{\beta} + \omega_l)} \right), \quad (5)$$

where  $d_i$  is the number of failures at time  $t_{(i)}$ . The residual for the  $l$ th observation is computed as:

$$r(t_l) = \hat{H}(t_l) \exp(-z_l^T \hat{\beta} + \omega_l)$$

where  $\hat{H}(t_l) = \hat{H}(t_{(i)}, t_{(i)} \leq t_l < t_{(i+1)})$ . The deviance is defined as  $-2 \times (\text{logarithm of marginal likelihood})$ . There are two ways to test whether individual covariates are significant: the differences between the deviances of nested models can be compared with the appropriate  $\chi^2$ -distribution; or, the asymptotic normality of the parameter estimates can be used to form  $z$  tests by dividing the estimates by their standard errors or the score function for the model under the null hypothesis can be used to form  $z$  tests.

## 4 References

- [1] Cox D R (1972) Regression models in life tables (with discussion) *J. Roy. Statist. Soc. Ser. B* **34** 187–220
- [2] Kalbfleisch J D and Prentice R L (1980) *The Statistical Analysis of Failure Time Data* Wiley
- [3] Gross A J and Clark V A (1975) *Survival Distributions: Reliability Applications in the Biomedical Sciences* Wiley

## 5 Parameters

1: OFFSET — CHARACTER\*1

*Input*

*On entry:* indicates if an offset is to be used.

If OFFSET = 'Y' then an offset must be included in OMEGA.

If OFFSET = 'N' no offset is included in the model.

*Constraint:* OFFSET = 'Y' or 'N'.

- 2:** N — INTEGER *Input*  
*On entry:* the number of data points,  $n$ .  
*Constraint:*  $N \geq 2$ .
- 3:** M — INTEGER *Input*  
*On entry:* the number of covariates in array Z.  
*Constraint:*  $M \geq 1$ .
- 4:** NS — INTEGER *Input*  
*On entry:* the number of strata. If  $NS > 0$  then the stratum for each observation must be supplied in ISI.  
*Constraint:*  $NS \geq 0$ .
- 5:** Z(LDZ,N) — *real* array *Input*  
*On entry:* the  $i$ th row must contain the covariates which are associated with the  $i$ th failure time given in T.
- 6:** LDZ — INTEGER *Input*  
*On entry:* the first dimension of the array Z as declared in the (sub)program from which G12BAF is called.  
*Constraint:*  $LDZ \geq N$ .
- 7:** ISZ(M) — INTEGER array *Input*  
*On entry:* indicates which subset of covariates is to be included in the model.  
 If  $ISZ(j) \geq 1$  the  $j$ th covariate is included in the model.  
 If  $ISZ(j) = 0$  the  $j$ th covariate is excluded from the model and not referenced.  
*Constraints:*  $ISZ(j) \geq 0$  and at least one and at most  $n_0 - 1$  elements of ISZ must be non-zero where  $n_0$  is the number of observations excluding any with zero value of ISI.
- 8:** IP — INTEGER *Input*  
*On entry:* the number of covariates included in the model as indicated by ISZ.  
*Constraint:* IP = number of non-zero values of ISZ.
- 9:** T(N) — *real* array *Input*  
*On entry:* the vector of  $n$  failure censoring times.
- 10:** IC(N) — INTEGER array *Input*  
*On entry:* the status of the individual at time  $t$  given in T.  
 IC( $i$ ) = 0 indicates that the  $i$ th individual has failed at time T( $i$ ).  
 IC( $i$ ) = 1 indicates that the  $i$ th individual has been censored at time T( $i$ ).  
*Constraint:* IC( $i$ ) = 0 or 1 for  $i = 1, 2, \dots, N$ .
- 11:** OMEGA(\*) — *real* array *Input*  
**Note:** the dimension of the array OMEGA must be at least N if OFFSET = 'Y' and 1 otherwise.  
*On entry:* if OFFSET = 'Y' the offset,  $\omega_i, i = 1, 2, \dots, N$ . Otherwise OMEGA is not referenced.

- 12:** ISI(\*) — INTEGER array *Input*  
**Note:** the dimension of the array ISI must be at least N if NS > 0 and 1 otherwise.  
*On entry:* if NS > 0 the stratum indicators which also allow data points to be excluded from the analysis. If NS = 0 ISI is not referenced.  
 ISI(*i*) = *k* indicates that the *i*th data point is in the *k*th stratum, where  $k = 1, 2, \dots, NS$ .  
 ISI(*i*) = 0 indicates that the *i*th data point is omitted from the analysis.  
*Constraints:* if NS > 0,  $0 \leq ISI(i) \leq NS$  for  $i = 1, 2, \dots, N$ , and more than IP values of  $ISI(i) > 0$ .
- 13:** DEV — *real* *Output*  
*On exit:* the deviance, that is  $-2 \times (\text{maximized log marginal likelihood})$ .
- 14:** B(IP) — *real* array *Input/Output*  
*On entry:* initial estimates of the covariate coefficient parameters  $\beta$ . B(*j*) must contain the initial estimate of the coefficient of the covariate in Z corresponding to the *j*th non-zero value of ISZ.  
*Suggested values:* In many cases an initial value of zero for B(*j*) may be used. For other suggestions see Section 8.  
*On exit:* B(*j*) contains the estimate  $\hat{\beta}_i$ , the coefficient of the covariate stored in the *i*th column of Z where *i* is the *j*th non-zero value in the array ISZ.
- 15:** SE(IP) — *real* array *Output*  
*On exit:* SE(*j*) is the asymptotic standard error of the estimate contained in B(*j*) and score function in SC(*j*) for  $j = 1, 2, \dots, IP$ .
- 16:** SC(IP) — *real* array *Output*  
*On exit:* SC(*j*) is the value of the score function,  $U_j(\beta)$ , for the estimate contained in B(*j*).
- 17:** COV(IP\*(IP+1)/2) — *real* array *Output*  
*On exit:* the variance-covariance matrix of the parameter estimates in B stored in packed form by column, i.e. the covariance between the parameter estimates given in B(*i*) and B(*j*),  $j \geq i$ , is stored in COV( $j(j-1)/2 + i$ ).
- 18:** RES(N) — *real* array *Output*  
*On exit:* the residuals,  $r(t_l), l = 1, 2, \dots, N$ .
- 19:** ND — INTEGER *Output*  
*On exit:* the number of distinct failure times.
- 20:** TP(NDMAX) — *real* array *Output*  
*On exit:* TP(*i*) contains the *i*th distinct failure time for  $i = 1, 2, \dots, ND$ .
- 21:** SUR(NDMAX,\*) — *real* array *Output*  
**Note:** the second dimension of the array SUR must be at least  $\max(NS, 1)$ .  
*On exit:*  
 if NS = 0 SUR(*i*, 1) contains the estimated survival function for the *i*th distinct failure time,  
 if NS > 0 SUR(*i*, *k*) contains the estimated survival function for the *i*th distinct failure time in the *k*th stratum.
- 22:** NDMAX — INTEGER *Input*  
*On entry:* the first dimension of the array SUR as declared in the (sub)program from which G12BAF is called.  
*Constraint:* NDMAX  $\geq$  the number of distinct failure times. This is returned in ND.

- 23:** TOL — *real* *Input*  
*On entry:* indicates the accuracy required for the estimation. Convergence is assumed when the decrease in deviance is less than  $TOL \times (1.0 + \text{CurrentDeviance})$ . This corresponds approximately to an absolute precision if the deviance is small and a relative precision if the deviance is large.  
*Constraint:*  $TOL \geq 10 \times \text{machine precision}$ .
- 24:** MAXIT — INTEGER *Input*  
*On entry:* the maximum number of iterations to be used for computing the estimates. If MAXIT is set to 0 then the standard errors, score functions, variance-covariance matrix and the survival function are computed for the input value of  $\beta$  in B but  $\beta$  is not updated.  
*Constraint:*  $MAXIT \geq 0$ .
- 25:** IPRINT — INTEGER *Input*  
*On entry:* indicates if the printing of information on the iterations is required. If  $IPRINT \leq 0$ , there is no printing, if  $IPRINT \geq 1$  then the deviance and the current estimates are printed every IPRINT iterations.  
 When printing occurs the output is directed to the current advisory message unit (see X04ABF).
- 26:** WK(IP\*(IP+9)/2+N) — *real* array *Workspace*
- 27:** IWK(2\*N) — INTEGER array *Workspace*
- 28:** IFAIL — INTEGER *Input/Output*  
*On entry:* IFAIL must be set to 0, -1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.  
*On exit:* IFAIL = 0 unless the routine detects an error (see Section 6).

## 6 Errors 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 detected by the routine:

IFAIL = 1

- On entry,  $OFFSET \neq 'Y'$  or  $'N'$ ,
- or  $M < 1$ ,
- or  $N < 2$ ,
- or  $NS < 0$ ,
- or  $LDZ < N$ ,
- or  $TOL < 10 \times \text{machine precision}$ ,
- or  $MAXIT < 0$ .

IFAIL = 2

- On entry,  $ISZ(i) < 0$  for some  $i$ ,
- or the value of IP is incompatible with ISZ,
- or  $IC(i) \neq 1$  or  $0$ .
- or  $ISI(i) < 0$  or  $ISI(i) > NS$ ,
- or number of values of  $ISZ(i) > 0$  is greater or equal to  $n_0$ , the number of observations excluding any with  $ISI(i) = 0$ ,
- or all observations are censored, i.e.,  $IC(i) = 1$  for all  $i$ ,

or NDMAX is too small.

IFAIL = 3

The matrix of second partial derivatives is singular. Try different starting values or include fewer covariates.

IFAIL = 4

Overflow has been detected. Try using different starting values.

IFAIL = 5

Convergence has not been achieved in MAXIT iterations. The progress toward convergence can be examined by using a non-zero value of IPRINT. Any non-convergence may be due to a linear combination of covariates being monotonic with time.

Full results are returned.

IFAIL = 6

In the current iteration 10 step halvings have been performed without decreasing the deviance from the previous iteration. Convergence is assumed.

## 7 Accuracy

The accuracy is specified by TOL.

## 8 Further Comments

The routine uses mean centering which involves subtracting the means from the covariables prior to computation of any statistics. This helps to minimize the effect of outlying observations and accelerates convergence.

If the initial estimates are poor then there may be a problem with overflow in calculating  $\exp(\beta^T z_i)$  or there may be non-convergence. Reasonable estimates can often be obtained by fitting an exponential model using G02GCF.

## 9 Example

The data are the remission times for two groups of leukemia patients (see Gross and Clark [3] p242). A dummy variable indicates which group they come from. An initial estimate is computed using the exponential model and then the Cox proportional hazard model is fitted and parameter estimates and the survival function are printed.

### 9.1 Program Text

**Note.** The listing of the example program presented below uses bold italicised terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*      G12BAF Example Program Text
*      Mark 17 Release. NAG Copyright 1995.
*
*      .. Parameters ..
INTEGER          NMAX, NDMAX, MMAX, SMAX, NIN, NOUT
PARAMETER       (NMAX=42, NDMAX=42, MMAX=2, SMAX=1, NIN=5, NOUT=6)
*      .. Local Scalars ..
real           DEV, TOL
INTEGER         I, IDF, IFAIL, IP, IPRINT, IRANK, J, LDZ, M,
+              MAXIT, N, ND, NS
*      .. Local Arrays ..
```

```

      real          B(MMAX), COV(MMAX*(MMAX+1)/2), OMEGA(NMAX),
+                RES(NMAX), SC(MMAX), SE(MMAX), SUR(NDMAX,SMAX),
+                T(NMAX), TP(NDMAX), V(NMAX,MMAX+7),
+                WK(MMAX*(MMAX+9)/2+NMAX), Y(NMAX), Z(NMAX,MMAX)
      INTEGER      IC(NMAX), ISI(NMAX), ISZ(MMAX), IWK(2*NMAX)
*
* .. External Subroutines ..
      EXTERNAL     GO2GCF, G12BAF
*
* .. Intrinsic Functions ..
      INTRINSIC   real, LOG, MAX
*
* .. Executable Statements ..
      WRITE (NOUT,*) 'G12BAF Example Program Results'
*
* Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, M, NS, MAXIT, IPRINT
      IF (N.LE.NMAX .AND. M.LE.MMAX) THEN
        IF (NS.GT.0) THEN
          DO 20 I = 1, N
            READ (NIN,*) T(I), (Z(I,J),J=1,M), IC(I), ISI(I)
20          CONTINUE
          ELSE
            DO 40 I = 1, N
              READ (NIN,*) T(I), (Z(I,J),J=1,M), IC(I)
40          CONTINUE
          END IF
          READ (NIN,*) (ISZ(I),I=1,M), IP
          LDZ = NMAX
          TOL = 0.00005e0
          DO 60 I = 1, N
            Y(I) = 1.0e0 - real(IC(I))
            V(I,7) = LOG(T(I))
60          CONTINUE
          IFAIL = -1
          CALL GO2GCF('L', 'M', 'Y', 'U', N, Z, LDZ, M, ISZ, IP+1, Y, RES, 0.0e0, DEV,
+                IDF, B, IRANK, SE, COV, V, NMAX, TOL, MAXIT, 0, 0.0e0, WK,
+                IFAIL)
          DO 80 I = 1, IP
            B(I) = B(I+1)
80          CONTINUE
          IF (IRANK.NE.IP+1) THEN
            WRITE (NOUT,*) ' WARNING: covariates not of full rank'
          END IF
          IFAIL = 0
*
*
          CALL G12BAF('No-offset', N, M, NS, Z, LDZ, ISZ, IP, T, IC, OMEGA, ISI, DEV,
+                B, SE, SC, COV, RES, ND, TP, SUR, NDMAX, TOL, MAXIT, IPRINT,
+                WK, IWK, IFAIL)
*
          WRITE (NOUT,*)
          WRITE (NOUT,*) ' Parameter      Estimate',
+                '      Standard Error'
          WRITE (NOUT,*)
          DO 100 I = 1, IP
            WRITE (NOUT,99999) I, B(I), SE(I)
100          CONTINUE
          WRITE (NOUT,*)
          WRITE (NOUT,99998) ' Deviance = ', DEV
          WRITE (NOUT,*)
          WRITE (NOUT,*) '      Time      Survivor Function'

```

```

        WRITE (NOUT,*)
        NS = MAX(NS,1)
        DO 120 I = 1, ND
            WRITE (NOUT,99997) TP(I), (SUR(I,J),J=1,NS)
120     CONTINUE
        END IF
        STOP
*
99999 FORMAT (I6,2(10X,F8.4))
99998 FORMAT (A,e13.4)
99997 FORMAT (F10.0,3(5X,F8.4))
        END

```

## 9.2 Program Data

G12BAF Example Program Data

42 1 0 20 0

```

1 0 0
1 0 0
2 0 0
2 0 0
3 0 0
4 0 0
4 0 0
5 0 0
5 0 0
8 0 0
8 0 0
8 0 0
8 0 0
11 0 0
11 0 0
12 0 0
12 0 0
15 0 0
17 0 0
22 0 0
23 0 0
6 1 0
6 1 0
6 1 0
7 1 0
10 1 0
13 1 0
16 1 0
22 1 0
23 1 0
6 1 1
9 1 1
10 1 1
11 1 1
17 1 1
19 1 1
20 1 1
25 1 1
32 1 1

```



```

32 1 1
34 1 1
35 1 1
   1  1

```

### 9.3 Program Results

G12BAF Example Program Results

Parameter	Estimate	Standard Error
1	-1.5091	0.4096

Deviance = 0.1728E+03

Time	Survivor Function
1.	0.9640
2.	0.9264
3.	0.9065
4.	0.8661
5.	0.8235
6.	0.7566
7.	0.7343
8.	0.6506
10.	0.6241
11.	0.5724
12.	0.5135
13.	0.4784
15.	0.4447
16.	0.4078
17.	0.3727
22.	0.2859
23.	0.1908

---