# NAG Library Routine Document

# G12BAF

**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, ISI,        &
                   DEV, B, SE, SC, COV, RES, ND, TP, SUR, NDMAX, TOL,           &
                   MAXIT, IPRINT, WK, IWK, IFAIL)

INTEGER            N, M, NS, LDZ, ISZ(M), IP, IC(N), ISI(*), ND, NDMAX,         &
                   MAXIT, IPRINT, IWK(2*N), IFAIL
REAL (KIND=nag_wp) Z(LDZ,M), T(N), OMEGA(*), DEV, B(IP), SE(IP), SC(IP),       &
                   COV(IP*(IP+1)/2), RES(N), TP(NDMAX), 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 $t_i$, for $i=1,2,\ldots,n$, be 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 (see Cox (1972)) $\lambda(t,z)$ is of the form:

$$\lambda(t,z)=\lambda_0(t)\exp\left(z^{\mathrm{T}}\beta+\omega\right)$$

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, $t_{(1)}<\cdots<t_{(n_d)}$ such that $d_i$ individuals fail at $t_{(i)}$, it follows that the marginal likelihood for $\beta$ is well approximated (see Kalbfleisch and Prentice (1980)) by:

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

where $s_i$ is the sum of the covariates of individuals observed to fail at $t_{(i)}$ and $R\left(t_{(i)}\right)$ 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:

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

for $j=1,2,\ldots,p$, where $s_{ji}$ is the $j$th element in the vector $s_i$ and

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

Similarly,

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

where

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

$U_j(\beta)$ 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 $I(\beta)^{-1} = \left[I_{hj}(\beta)\right]^{-1}$ 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 $n_k$, for $k = 1, 2, \ldots, \nu$ with $n = \sum_{k=1}^{\nu} n_k$, then rather than maximizing (1) to obtain $\hat{\beta}$, the following marginal likelihood is maximized:

$$L = \prod_{k=1}^{\nu} L_k, \tag{4}$$

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 $\exp\left(-\hat{H}\left(t_{(i)}\right)\right)$, where

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

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

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

where $\hat{H}(t_l) = \hat{H}\left(t_{(i)}\right), t_{(i)} \leq t_l < t_{(i+1)}$. The deviance is defined as $-2 \times$ (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

Cox D R (1972) Regression models in life tables (with discussion) *J. Roy. Statist. Soc. Ser. B* **34** 187–220

Gross A J and Clark V A (1975) *Survival Distributions: Reliability Applications in the Biomedical Sciences* Wiley

Kalbfleisch J D and Prentice R L (1980) *The Statistical Analysis of Failure Time Data* Wiley

## 5 Parameters

1: OFFSET – CHARACTER(1) *Input*

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

OFFSET = 'Y'
An offset must be included in OMEGA.

OFFSET = 'N'
No offset is included in the model.

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

2: N – INTEGER *Input*

*On entry*: $n$, the number of data points.

*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,M) – REAL (KIND=nag_wp) 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.

ISZ($j$) $\geq$ 1
The $j$th covariate is included in the model.

ISZ($j$) = 0
The $j$th covariate is excluded from the model and not referenced.

*Constraint*: ISZ($j$) $\geq$ 0 and at least one and at most $n_0 - 1$ elements of ISZ must be nonzero 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.

*Constraints*:

IP $\geq$ 1;
IP = number of nonzero values of ISZ.

9: T(N) – REAL (KIND=nag_wp) 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$(i) = 0$
The $i$th individual has failed at time T$(i)$.

IC$(i) = 1$
The $i$th individual has been censored at time T$(i)$.

*Constraint*: IC$(i) = 0$ or 1, for $i = 1, 2, \ldots, \mathrm{N}$.

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

**Note**: the dimension of the array OMEGA must be at least N if OFFSET = 'Y', and at least 1 otherwise.

*On entry*: if OFFSET = 'Y', the offset, $\omega_i$, for $i = 1, 2, \ldots, \mathrm{N}$. 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 at least 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$(i) = k$
The $i$th data point is in the $k$th stratum, where $k = 1, 2, \ldots, \mathrm{NS}$.

ISI$(i) = 0$
The $i$th data point is omitted from the analysis.

*Constraint*: if NS $> 0$, $0 \leq$ ISI$(i) \leq$ NS and more than IP values of ISI$(i) > 0$, for $i = 1, 2, \ldots, \mathrm{N}$.

13: DEV – REAL (KIND=nag_wp) *Output*

*On exit*: the deviance, that is $-2 \times$ (maximized log marginal likelihood).

14: B(IP) – REAL (KIND=nag_wp) 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 nonzero value of ISZ.

*Suggested value*: 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 nonzero value in the array ISZ.

15: SE(IP) – REAL (KIND=nag_wp) 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 $j = 1, 2, \ldots, \mathrm{IP}$.

16: SC(IP) – REAL (KIND=nag_wp) 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 (KIND=nag_wp) 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 (KIND=nag_wp) array *Output*

*On exit*: the residuals, $r(t_l)$, for $l = 1, 2, \ldots, \text{N}$.

19: ND – INTEGER *Output*

*On exit*: the number of distinct failure times.

20: TP(NDMAX) – REAL (KIND=nag_wp) array *Output*

*On exit*: TP$(i)$ contains the $i$th distinct failure time, for $i = 1, 2, \ldots, \text{ND}$.

21: SUR(NDMAX,$*$) – REAL (KIND=nag_wp) array *Output*

**Note**: the second dimension of the array SUR must be at least $\max(\text{NS}, 1)$.

*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 dimension of the array TP and the first dimension of the array SUR as declared in the (sub)program from which G12BAF is called.

*Constraint*: NDMAX ≥ the number of distinct failure times. This is returned in ND.

23: TOL – REAL (KIND=nag_wp) *Input*

*On entry*: indicates the accuracy required for the estimation. Convergence is assumed when the decrease in deviance is less than TOL × (1.0 + CurrentDeviance). This corresponds approximately to an absolute precision if the deviance is small and a relative precision if the deviance is large.

*Constraint*: TOL ≥ 10 × ***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 ≥ 0.

25: IPRINT – INTEGER *Input*

*On entry*: indicates if the printing of information on the iterations is required.

IPRINT ≤ 0
No printing.

IPRINT ≥ 1
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 (KIND=nag_wp) 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. 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, OFFSET $\neq$ 'Y' or 'N',
or M $< 1$,
or N $< 2$,
or NS $< 0$,
or IP $< 1$,
or LDZ $<$ N,
or TOL $< 10 \times$ ***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 than 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 nonzero 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

G12BAF 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 $\exp(\beta^{\mathrm{T}} z_i)$ 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 page 242 of Gross and Clark (1975)). 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

```
    Program g12bafe

!     G12BAF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: g02gcf, g12baf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: dev, tol
      Integer                          :: i, idf, ifail, ip, ip1, iprint,      &
                                          irank, ldv, ldz, lisi, lomega, m,    &
                                          maxit, n, nd, ndmax, ns
      Character (1)                    :: offset
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: b(:), cov(:), omega(:), res(:),      &
                                          sc(:), se(:), sur(:,:), t(:), tp(:), &
                                          v(:,:), wk(:), y(:), z(:,:)
      Integer, Allocatable             :: ic(:), isi(:), isz(:), iwk(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: count, eoshift, log, max, real
!     .. Executable Statements ..
      Write (nout,*) 'G12BAF Example Program Results'
      Write (nout,*)

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

!     Read in problem size
      Read (nin,*) n, m, ns, maxit, iprint, offset

      If (offset=='Y' .Or. offset=='y') Then
        lomega = n
      Else
        lomega = 0
      End If
      If (ns>0) Then
        lisi = n
      Else
        lisi = 0
      End If
      ldz = n
      ndmax = n
      ldv = n
      Allocate (z(ldz,m),isz(m),t(n),ic(n),omega(lomega),isi(lisi),res(n), &
        tp(ndmax),sur(ndmax,max(ns,1)),iwk(2*n),y(n))

!     Read in the data
      If (ns>0) Then
        If (lomega==0) Then
          Read (nin,*)(t(i),z(i,1:m),ic(i),isi(i),i=1,n)
        Else
          Read (nin,*)(t(i),z(i,1:m),ic(i),isi(i),omega(i),i=1,n)
        End If
      Else
        If (lomega==0) Then
          Read (nin,*)(t(i),z(i,1:m),ic(i),i=1,n)
        Else
          Read (nin,*)(t(i),z(i,1:m),ic(i),omega(i),i=1,n)
        End If
      End If

!     Read in the variable indication
      Read (nin,*) isz(1:m)

!     Calculate number of parameters in the model
      ip = count(isz(1:m)>0)

!     We are fitting a mean in the exponential model, so arrays used by G02GCF
!     need to be based on IP + 1
      ip1 = ip + 1
      Allocate (b(ip1),se(ip1),sc(ip),cov(ip1*(ip1+1)/2),wk(ip1*(ip1+ &
        9)/2+n),v(ldv,ip1+7))

!     Specifiy tolerance to use
      tol = 5.0E-5_nag_wp

!     Get initial estimates by fitting an exponential model
      Do i = 1, n
        y(i) = 1.0E0_nag_wp - real(ic(i),kind=nag_wp)
        v(i,7) = log(t(i))
      End Do

!     Fit exponential model
      ifail = -1
      Call g02gcf('L','M','Y','U',n,z,ldz,m,isz,ip1,y,res,0.0E0_nag_wp,dev, &
        idf,b,irank,se,cov,v,ldv,tol,maxit,0,0.0E0_nag_wp,wk,ifail)
      If (ifail/=0) Then
        If (ifail<5) Then
          Go To 100
        End If
      End If

!     Check exponential model was of full rank
      If (irank/=ip1) Then
        Write (nout,*) ' WARNING: covariates not of full rank'
      End If

!     Move all parameter estimates down one so as to drop the parameter
!     estimate for the mean.
      b = eoshift(b,1)

!     Fit Cox proportional hazards model
      ifail = -1
      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)
      If (ifail/=0) Then
        If (ifail<5) Then
          Go To 100
        End If
      End If

!     Display results
      Write (nout,*) ' Parameter      Estimate', '       Standard Error'
      Write (nout,*)
      Write (nout,99999)(i,b(i),se(i),i=1,ip)
      Write (nout,*)
      Write (nout,99998) ' Deviance = ', dev
      Write (nout,*)
      Write (nout,*) '    Time     Survivor Function'
      Write (nout,*)
      ns = max(ns,1)
      Write (nout,99997)(tp(i),sur(i,1:ns),i=1,nd)

100   Continue

99999 Format (I6,10X,F8.4,10X,F8.4)
99998 Format (A,E13.4)
99997 Format (F10.0,5X,F8.4)
    End Program g12bafe
```

## 9.2 Program Data

```
G12BAF Example Program Data
42 1 0 20 0 'N'   : N,M,NS,MAXIT,IPRINT,OFFSET
 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           : T,Z,IC
   1             : ISZ
```

### 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
```