NAG Library Routine Document

G11BAF

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

G11BAF computes a table from a set of classification factors using a selected statistic.

2 Specification

```
SUBROUTINE G11BAF (STAT, UPDATE, WEIGHT, N, NFAC, ISF, LFAC, IFAC, LDF,     &
                   Y, WT, TABLE, MAXT, NCELLS, NDIM, IDIM, ICOUNT, AUXT,    &
                   IWK, IFAIL)
INTEGER            N, NFAC, ISF(NFAC), LFAC(NFAC), IFAC(LDF,NFAC),          &
                   LDF, MAXT, NCELLS, NDIM, IDIM(NFAC), ICOUNT(MAXT),       &
                   IWK(2*NFAC), IFAIL
REAL (KIND=nag_wp) Y(N), WT(*), TABLE(MAXT), AUXT(*)
CHARACTER(1)       STAT, UPDATE, WEIGHT
```

3 Description

A dataset may include both classification variables and general variables. The classification variables, known as factors, take a small number of values known as levels. For example, the factor sex would have the levels male and female. These can be coded as 1 and 2 respectively. Given several factors, a multi-way table can be constructed such that each cell of the table represents one level from each factor. For example, the two factors sex and habitat, habitat having three levels (inner-city, suburban and rural) define the 2×3 contingency table

Sex	**Habitat**		
	Inner-city	Suburban	Rural
Male			
Female			

For each cell statistics can be computed. If a third variable in the dataset was age, then for each cell the average age could be computed:

Sex	**Habitat**		
	Inner-city	Suburban	Rural
Male	25.5	30.3	35.6
Female	23.2	29.1	30.4

That is the average age for all observations for males living in rural areas is 35.6. Other statistics can also be computed: the number of observations, the total, the variance, the largest value and the smallest value.

G11BAF computes a table for one of the selected statistics. The factors have to be coded with levels $1, 2, \ldots$. Weights can be used to eliminate values from the calculations, e.g., if they represent 'missing values'. There is also the facility to update an existing table with the addition of new observations.

4 References

John J A and Quenouille M H (1977) *Experiments: Design and Analysis* Griffin

Kendall M G and Stuart A (1969) *The Advanced Theory of Statistics (Volume 1)* (3rd Edition) Griffin

West D H D (1979) Updating mean and variance estimates: An improved method *Comm. ACM* **22** 532–555

5 Arguments

1: STAT – CHARACTER(1) *Input*

On entry: indicates which statistic is to be computed for the table cells.

STAT = 'N'
The number of observations for each cell.

STAT = 'T'
The total for the variable in Y for each cell.

STAT = 'A'
The average (mean) for the variable in Y for each cell.

STAT = 'V'
The variance for the variable in Y for each cell.

STAT = 'L'
The largest value for the variable in Y for each cell.

STAT = 'S'
The smallest value for the variable in Y for each cell.

Constraint: STAT = 'N', 'T', 'A', 'V', 'L' or 'S'.

2: UPDATE – CHARACTER(1) *Input*

On entry: indicates if an existing table is to be updated by further observation.

UPDATE = 'I'
The table cells will be initialized to zero before tabulations take place.

UPDATE = 'U'
The table input in TABLE will be updated. The arguments NCELLS, TABLE, ICOUNT and AUXT must remain unchanged from the previous call to G11BAF.

Constraint: UPDATE = 'I' or 'U'.

3: WEIGHT – CHARACTER(1) *Input*

On entry: indicates if weights are to be used.

WEIGHT = 'U'
Weights are not used and unit weights are assumed.

WEIGHT = 'W' or 'V'
Weights are used and must be supplied in WT. The only difference between WEIGHT = 'W' and WEIGHT = 'V' is if the variance is computed.

WEIGHT = 'W'
The divisor for the variance is the sum of the weights minus one and if WEIGHT = 'V', the divisor is the number of observations with nonzero weights minus one. The former is useful if the weights represent the frequency of the observed values.

If STAT = 'T' or 'A', the weighted total or mean is computed respectively.

If STAT = 'N', 'L' or 'S', the only effect of weights is to eliminate values with zero weights from the computations.

Constraint: WEIGHT = 'U', 'V' or 'W'.

4: N – INTEGER *Input*

On entry: the number of observations.

Constraint: $\mathrm{N} \geq 2$.

5: NFAC – INTEGER *Input*

On entry: the number of classifying factors in IFAC.

Constraint: $\mathrm{NFAC} \geq 1$.

6: ISF(NFAC) – INTEGER array *Input*

On entry: indicates which factors in IFAC are to be used in the tabulation.

If $\mathrm{ISF}(i) > 0$ the ith factor in IFAC is included in the tabulation.

Note that if $\mathrm{ISF}(i) \leq 0$, for $i = 1, 2, \ldots, \mathrm{NFAC}$ then the statistic for the whole sample is calculated and returned in a 1×1 table.

7: LFAC(NFAC) – INTEGER array *Input*

On entry: the number of levels of the classifying factors in IFAC.

Constraint: if $\mathrm{ISF}(i) > 0$, $\mathrm{LFAC}(i) \geq 2$, for $i = \mathrm{Ai}, \ldots, \mathrm{Ai}$.

8: IFAC(LDF, NFAC) – INTEGER array *Input*

On entry: the NFAC coded classification factors for the N observations.

Constraint: $1 \leq \mathrm{IFAC}(i, j) \leq \mathrm{LFAC}(j)$, for $i = 1, 2, \ldots, \mathrm{N}$ and $j = 1, 2, \ldots, \mathrm{NFAC}$.

9: LDF – INTEGER *Input*

On entry: the first dimension of the array IFAC as declared in the (sub)program from which G11BAF is called.

Constraint: $\mathrm{LDF} \geq \mathrm{N}$.

10: Y(N) – REAL (KIND=nag_wp) array *Input*

On entry: the variable to be tabulated. If STAT = 'N', Y is not referenced.

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

Note: the dimension of the array WT must be at least N if WEIGHT = 'W' or 'V', and at least 1 otherwise.

On entry: if WEIGHT = 'W' or 'V', WT must contain the N weights. Otherwise WT is not referenced.

Constraint: if WEIGHT = 'W' or 'V', $\mathrm{WT}(i) \geq 0.0$, for $i = \mathrm{Ai}, \ldots, \mathrm{Ai}$.

12: TABLE(MAXT) – REAL (KIND=nag_wp) array *Input/Output*

On entry: if UPDATE = 'U', TABLE must be unchanged from the previous call to G11BAF, otherwise TABLE need not be set.

On exit: the computed table. The NCELLS cells of the table are stored so that for any two factors the index relating to the factor referred to later in LFAC and IFAC changes faster. For further details see Section 9.

13: MAXT – INTEGER *Input*

On entry: the maximum size of the table to be computed.

Constraint: MAXT $\geq$ product of the levels of the factors included in the tabulation.

14: NCELLS – INTEGER *Input/Output*

On entry: if UPDATE = 'U', NCELLS must be unchanged from the previous call to G11BAF, otherwise NCELLS need not be set.

On exit: the number of cells in the table.

15: NDIM – INTEGER *Output*

On exit: the number of factors defining the table.

16: IDIM(NFAC) – INTEGER array *Output*

On exit: the first NDIM elements contain the number of levels for the factors defining the table.

17: ICOUNT(MAXT) – INTEGER array *Input/Output*

On entry: if UPDATE = 'U', ICOUNT must be unchanged from the previous call to G11BAF, otherwise ICOUNT need not be set.

On exit: a table containing the number of observations contributing to each cell of the table, stored identically to TABLE. Note if STAT = 'N' this is the same as is returned in TABLE.

18: AUXT($*$) – REAL (KIND=nag_wp) array *Input/Output*

Note: the dimension of the array AUXT must be at least NCELLS if STAT = 'A', $2 \times$ NCELLS if STAT = 'V', and at least 1 otherwise.

On entry: if UPDATE = 'U', AUXT must be unchanged from the previous call to G11BAF, otherwise AUXT need not be set.

On exit: if STAT = 'A' or 'V', the first NCELLS values hold the table containing the sum of the weights for the observations contributing to each cell, stored identically to TABLE.

If STAT = 'V', the second set of NCELLS values hold the table of cell means. Otherwise AUXT is not referenced.

19: IWK($2 \times$ NFAC) – INTEGER array *Workspace*

20: 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, if you are not familiar with this argument, 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, N < 2,
or NFAC < 1,
or LDF $<$ N,
or UPDATE $\neq$ 'I' or 'U',
or WEIGHT $\neq$ 'U', 'W' or 'V',
or STAT $\neq$ 'N', 'T', 'A', 'V', 'L' or 'S'.

IFAIL $= 2$

On entry, ISF$(i) > 0$ and LFAC$(i) < 2$, for some i,
or IFAC$(i, j) < 1$, for some i, j,
or IFAC$(i, j) >$ LFAC(j) for some i, j,
or MAXT is too small,
or WEIGHT $=$ 'W' or 'V' and WT$(i) < 0.0$, for some i.

IFAIL $= 3$

STAT $=$ 'V' and the divisor for the variance is ≤ 0.0.

IFAIL $= 4$

UPDATE $=$ 'U' and at least one of NCELLS, TABLE, AUXT or ICOUNT have been changed since previous call to G11BAF.

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

Only applicable when STAT $=$ 'V'. In this case a one pass algorithm is used as described by West (1979).

8 Parallelism and Performance

G11BAF is not threaded in any implementation.

9 Further Comments

The tables created by G11BAF and stored in TABLE, ICOUNT and, depending on STAT, also in AUXT are stored in the following way. Let there be n factors defining the table with factor k having l_k levels, then the cell defined by the levels i_1, $i_2, \ldots, i_n$ of the factors is stored in the mth cell given by

$$m = 1 + \sum_{k=1}^{n} [(i_k - 1)c_k],$$

where $c_j = \prod_{k=j+1}^{n} l_k$, for $j = 1, 2, \ldots, n-1$ and $c_n = 1$.

10 Example

The data, given by John and Quenouille (1977), is for a 3×6 factorial experiment in 3 blocks of 18 units. The data is input in the order, blocks, factor with 3 levels, factor with 6 levels, yield. The 3×6 table of treatment means for yield over blocks is computed and printed.

10.1 Program Text

```
    Program g11bafe

!     G11BAF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: g11baf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, j, k, lauxt, ldf, lwt,     &
                                          maxt, n, ncells, ncol, ndim, nfac,   &
                                          nrow
      Character (1)                    :: stat, weight
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: auxt(:), table(:), wt(:), y(:)
      Integer, Allocatable             :: icount(:), idim(:), ifac(:,:),       &
                                          isf(:), iwk(:), lfac(:)
!     .. Executable Statements ..
      Write (nout,*) 'G11BAF Example Program Results'
      Write (nout,*)

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

!     Read in the problem size
      Read (nin,*) stat, weight, n, nfac

      If (weight=='W' .Or. weight=='w' .Or. weight=='V' .Or. weight=='v') Then
        lwt = n
      Else
        lwt = 0
      End If
      ldf = n
      Allocate (isf(nfac),lfac(nfac),ifac(ldf,nfac),y(n),wt(lwt),idim(nfac),   &
        iwk(2*nfac))

!     Read in data
      If (lwt>0) Then
        Read (nin,*)(ifac(i,1:nfac),y(i),wt(i),i=1,n)
      Else
        Read (nin,*)(ifac(i,1:nfac),y(i),i=1,n)
      End If
```

```
      Read (nin,*) lfac(1:nfac)
      Read (nin,*) isf(1:nfac)

!     Calculate MAXT
      maxt = 1
      Do i = 1, nfac
        If (isf(i)>0) Then
          maxt = maxt*lfac(i)
        End If
      End Do

      Select Case (stat)
      Case ('A','a')
        lauxt = maxt
      Case ('V','v')
        lauxt = 2*maxt
      Case Default
        lauxt = 0
      End Select

      Allocate (table(maxt),icount(maxt),auxt(lauxt))

!     Compute table
      ifail = 0
      Call g11baf(stat,'I',weight,n,nfac,isf,lfac,ifac,ldf,y,wt,table,maxt,    &
        ncells,ndim,idim,icount,auxt,iwk,ifail)

!     Display results
      Write (nout,*) ' TABLE'
      Write (nout,*)
      ncol = idim(ndim)
      nrow = ncells/ncol
      k = 1
      Do i = 1, nrow
        Write (nout,99999)(table(j),'(',icount(j),')',j=k,k+ncol-1)
        k = k + ncol
      End Do

99999 Format (1X,6(F8.2,A,I2,A))
    End Program g11bafe
```

10.2 Program Data

```
G11BAF Example Program Data
'A' 'U' 54 3
1 1 1 274
1 2 1 361
1 3 1 253
1 1 2 325
1 2 2 317
1 3 2 339
1 1 3 326
1 2 3 402
1 3 3 336
1 1 4 379
1 2 4 345
1 3 4 361
1 1 5 352
1 2 5 334
1 3 5 318
1 1 6 339
1 2 6 393
1 3 6 358
2 1 1 350
2 2 1 340
2 3 1 203
2 1 2 397
2 2 2 356
2 3 2 298
2 1 3 382
```

```
2 2 3 376
2 3 3 355
2 1 4 418
2 2 4 387
2 3 4 379
2 1 5 432
2 2 5 339
2 3 5 293
2 1 6 322
2 2 6 417
2 3 6 342
3 1 1  82
3 2 1 297
3 3 1 133
3 1 2 306
3 2 2 352
3 3 2 361
3 1 3 220
3 2 3 333
3 3 3 270
3 1 4 388
3 2 4 379
3 3 4 274
3 1 5 336
3 2 5 307
3 3 5 266
3 1 6 389
3 2 6 333
3 3 6 353
3 3 6
0 1 1
```

10.3 Program Results

```
 G11BAF Example Program Results

  TABLE

   235.33( 3)  342.67( 3)  309.33( 3)  395.00( 3)  373.33( 3)  350.00( 3)
   332.67( 3)  341.67( 3)  370.33( 3)  370.33( 3)  326.67( 3)  381.00( 3)
   196.33( 3)  332.67( 3)  320.33( 3)  338.00( 3)  292.33( 3)  351.00( 3)
```