NAG Library Routine Document

G04BCF

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

G04BCF computes the analysis of variance for a general row and column design together with the treatment means and standard errors.

2   Specification

```
SUBROUTINE G04BCF (NREP, NROW, NCOL, Y, NT, IT, GMEAN, TMEAN, TABL,      &
                   LDTABL, C, LDC, IREP, RPMEAN, RMEAN, CMEAN, R, EF,    &
                   TOL, IRDF, WK, IFAIL)

INTEGER            NREP, NROW, NCOL, NT, IT(*), LDTABL, LDC, IREP(NT),   &
                   IRDF, IFAIL
REAL (KIND=nag_wp) Y(NREP*NROW*NCOL), GMEAN, TMEAN(NT),                  &
                   TABL(LDTABL,5), C(LDC,NT), RPMEAN(NREP),              &
                   RMEAN(NREP*NROW), CMEAN(NREP*NCOL),                   &
                   R(NREP*NROW*NCOL), EF(NT), TOL, WK(3*NT)
```

3   Description

In a row and column design the experimental material can be characterised by a two-way classification, nominally called rows and columns. Each experimental unit can be considered as being located in a particular row and column. It is assumed that all rows are of the same length and all columns are of the same length. Sets of equal numbers of rows/columns can be grouped together to form replicates, sometimes known as squares or rectangles, as appropriate.

If for a replicate, the number of rows, the number of columns and the number of treatments are equal and every treatment occurs once in each row and each column then the design is a Latin square. If this is not the case the treatments will be non-orthogonal to rows and columns. For example in the case of a lattice square each treatment occurs only once in each square.

For a row and column design, with t treatments in r rows and c columns and b replicates or squares with $n = brc$ observations the linear model is:

$$y_{ijk(l)} = \mu + \beta_i + \rho_j + \gamma_k + \tau_l + e_{ijk}$$

for $i = 1, 2 \ldots, b$, $j = 1, 2, \ldots, r$, $k = 1, 2 \ldots, c$ and $l = 1, 2, \ldots, t$, where β_i is the effect of the ith replicate, ρ_j is the effect of the jth row, γ_k is the effect of the kth column and the $ijk(l)$ notation indicates that the lth treatment is applied to the unit in row j, column k of replicate i.

To compute the analysis of variance for a row and column design the mean is computed and subtracted from the observations to give, $y'_{ijk(l)} = y_{ijk(l)} - \hat{\mu}$. Since the replicates, rows and columns are orthogonal the estimated effects, ignoring treatment effects, $\hat{\beta}_i$, $\hat{\rho}_j$, $\hat{\gamma}_k$, can be computed using the appropriate means of the $y'_{ijk(l)}$, and the unadjusted sum of squares computed as the appropriate sum of squared totals for the $y'_{ijk(l)}$ divided by number of units per total. The observations adjusted for replicates, rows and columns can then be computed by subtracting the estimated effects from $y'_{ijk(l)}$ to give $y''_{ijk(l)}$.

In the case of a Latin square design the treatments are orthogonal to replicates, rows and columns and so the treatment effects, $\hat{\tau}_l$, can be estimated as the treatment means of the adjusted observations, $y''_{ijk(l)}$. The treatment sum of squares is computed as the sum of squared treatment totals of the $y''_{ij(l)}$ divided by the number of times each treatment is replicated. Finally the residuals, and hence the residual sum of squares, are given by $r_{ij(l)} = y''_{ij(l)} - \hat{\tau}_l$.

For a design which is not orthogonal, for example a lattice square or an incomplete Latin square, the treatment effects adjusted for replicates, rows and columns need to be computed. The adjusted treatment effects are found as the solution to the equations:

$$A\hat{\tau} = \left(R - N_b N_b^{\mathrm{T}}/(rc) - N_r N_r^{\mathrm{T}}/(bc) - N_c N_c^{\mathrm{T}}/(br)\right)\hat{\tau} = q$$

where q is the vector of the treatment totals of the observations adjusted for replicates, rows and columns, $y''_{ijk(l)}$, R is a diagonal matrix with R_{ll} equal to the number of times the lth treatment is replicated, and N_b is the t by b incidence matrix, with $N_{l,i}$ equal to the number of times treatment l occurs in replicate i, with N_r and N_c being similarly defined for rows and columns. The solution to the equations can be written as:

$$\hat{\tau} = \Omega q$$

where, Ω is a generalized inverse of A. The solution is found from the eigenvalue decomposition of A. The residuals are first calculated by subtracting the estimated adjusted treatment effects from the adjusted observations to give $r'_{ij(l)} = y''_{ij(l)} - \hat{\tau}_l$. However, since only the unadjusted replicate, row and column effects have been removed and they are not orthogonal to treatments, the replicate, row and column means of the $r'_{ij(l)}$ have to be subtracted to give the correct residuals, $r_{ij(l)}$ and residual sum of squares.

Given the sums of squares, the mean squares are computed as the sums of squares divided by the degrees of freedom. The degrees of freedom for the unadjusted replicates, rows and columns are $b - 1$, $r - 1$ and $c - 1$ respectively and for the Latin square designs the degrees of freedom for the treatments is $t - 1$. In the general case the degrees of freedom for treatments is the rank of the matrix Ω. The F-statistic given by the ratio of the treatment mean square to the residual mean square tests the hypothesis:

$$H_0 : \tau_1 = \tau_2 = \cdots = \tau_t = 0.$$

The standard errors for the difference in treatment effects, or treatment means, for Latin square designs, are given by:

$$se\left(\hat{\tau}_j - \hat{\tau}_{j*}\right) = \sqrt{2s^2/(bt)}$$

where s^2 is the residual mean square. In the general case the variances of the treatment effects are given by:

$$\mathrm{Var}(\hat{\tau}) = \Omega s^2$$

from which the appropriate standard errors of the difference between treatment effects or the difference between adjusted means can be calculated.

The analysis of a row-column design can be considered as consisting of different strata: the replicate stratum, the rows within replicate and the columns within replicate strata and the units stratum. In the Latin square design all the information on the treatment effects is given at the units stratum. In other designs there may be a loss of information due to the non-orthogonality of treatments and replicates, rows and columns and information on treatments may be available in higher strata. The efficiency of the estimation at the units stratum is given by the (canonical) efficiency factors, these are the nonzero eigenvalues of the matrix, A, divided by the number of replicates in the case of equal replication, or by the mean of the number of replicates in the unequally replicated case, see John (1987). If more than one eigenvalue is zero then the design is said to be disconnected and information on some treatment comparisons can only be obtained from higher strata.

4   References

Cochran W G and Cox G M (1957) *Experimental Designs* Wiley

Davis O L (1978) *The Design and Analysis of Industrial Experiments* Longman

John J A (1987) *Cyclic Designs* Chapman and Hall

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

Searle S R (1971) *Linear Models* Wiley

5 Parameters

1: NREP – INTEGER *Input*

On entry: b, the number of replicates.

Constraint: NREP ≥ 1.

2: NROW – INTEGER *Input*

On entry: r, the number of rows per replicate.

Constraint: NROW ≥ 2.

3: NCOL – INTEGER *Input*

On entry: c, the number of columns per replicate.

Constraint: NCOL ≥ 2.

4: Y(NREP × NROW × NCOL) – REAL (KIND=nag_wp) array *Input*

On entry: the $n = brc$ observations ordered by columns within rows within replicates. That is $Y(rc(i-1) + r(j-1) + k)$ contains the observation from the kth column of the jth row of the ith replicate, for $i = 1, 2, \ldots, b$, $j = 1, 2, \ldots, r$ and $k = 1, 2, \ldots, c$.

5: NT – INTEGER *Input*

On entry: the number of treatments. If only replicates, rows and columns are required in the analysis then set NT $= 1$.

Constraint: NT ≥ 1.

6: IT(∗) – INTEGER array *Input*

Note: the dimension of the array IT must be at least NREP × NROW × NCOL if NT > 1, and at least 1 otherwise.

On entry: if NT > 1, IT(i) indicates which of the NT treatments unit i received, for $i = 1, 2, \ldots, n$.

If NT $= 1$, IT is not referenced.

Constraint: if NT ≥ 2, $1 \leq$ IT$(i) \leq$ NT, for $i = 1, 2, \ldots, n$.

7: GMEAN – REAL (KIND=nag_wp) *Output*

On exit: the grand mean, $\hat{\mu}$.

8: TMEAN(NT) – REAL (KIND=nag_wp) array *Output*

On exit: if NT ≥ 2, TMEAN(l) contains the (adjusted) mean for the lth treatment, $\hat{\mu}^* + \hat{\tau}_l$, for $l = 1, 2, \ldots, t$, where $\hat{\mu}^*$ is the mean of the treatment adjusted observations $y_{ijk(l)} - \hat{\tau}_l$. Otherwise TMEAN is not referenced.

9: TABL(LDTABL, 5) – REAL (KIND=nag_wp) array *Output*

On exit: the analysis of variance table. Column 1 contains the degrees of freedom, column 2 the sum of squares, and where appropriate, column 3 the mean squares, column 4 the F-statistic and column 5 the significance level of the F-statistic. Row 1 is for replicates, row 2 for rows, row 3 for columns, row 4 for treatments (if NT > 1), row 5 for residual and row 6 for total. Mean squares are computed for all but the total row, F-statistics and significance are computed for treatments, replicates, rows and columns. Any unfilled cells are set to zero.

10: LDTABL – INTEGER *Input*

On entry: the first dimension of the array TABL as declared in the (sub)program from which G04BCF is called.

Constraint: LDTABL ≥ 6.

11: C(LDC, NT) – REAL (KIND=nag_wp) array *Output*

On exit: the upper triangular part of C contains the variance-covariance matrix of the treatment effects, the strictly lower triangular part contains the standard errors of the difference between two treatment effects (means), i.e., $C(i, j)$ contains the covariance of treatment i and j if $j \geq i$ and the standard error of the difference between treatment i and j if $j < i$, for $i = 1, 2, \ldots, t$ and $j = 1, 2, \ldots, t$.

12: LDC – INTEGER *Input*

On entry: the first dimension of the array C as declared in the (sub)program from which G04BCF is called.

Constraint: LDC $\geq$ NT.

13: IREP(NT) – INTEGER array *Output*

On exit: if NT > 1, the treatment replications, R_{ll}, for $l = 1, 2, \ldots, \text{NT}$. Otherwise IREP is not referenced.

14: RPMEAN(NREP) – REAL (KIND=nag_wp) array *Output*

On exit: if NREP > 1, RPMEAN(i) contains the mean for the ith replicate, $\hat{\mu} + \hat{\beta}_i$, for $i = 1, 2, \ldots, b$. Otherwise RPMEAN is not referenced.

15: RMEAN(NREP × NROW) – REAL (KIND=nag_wp) array *Output*

On exit: RMEAN(j) contains the mean for the jth row, $\hat{\mu} + \hat{\rho}_i$, for $j = 1, 2, \ldots, r$.

16: CMEAN(NREP × NCOL) – REAL (KIND=nag_wp) array *Output*

On exit: CMEAN(k) contains the mean for the kth column, $\hat{\mu} + \hat{\gamma}_k$, for $k = 1, 2, \ldots, c$.

17: R(NREP × NROW × NCOL) – REAL (KIND=nag_wp) array *Output*

On exit: the residuals, r_i, for $i = 1, 2, \ldots, n$.

18: EF(NT) – REAL (KIND=nag_wp) array *Output*

On exit: if NT ≥ 2, the canonical efficiency factors. Otherwise EF is not referenced.

19: TOL – REAL (KIND=nag_wp) *Input*

On entry: the tolerance value used to check for zero eigenvalues of the matrix Ω. If TOL $= 0.0$ a default value of 0.00001 is used.

Constraint: TOL ≥ 0.0.

20: IRDF – INTEGER *Input*

On entry: an adjustment to the degrees of freedom for the residual and total.

IRDF ≥ 1
    The degrees of freedom for the total is set to $n - \text{IRDF}$ and the residual degrees of freedom adjusted accordingly.

IRDF = 0

the total degrees of freedom for the total is set to $n - 1$, as usual.

Constraint: IRDF $\geq$ 0.

21:   WK($3 \times$ NT) – REAL (KIND=nag_wp) array   *Workspace*

22:   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: G04BCF 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, NREP < 1,
or   NROW < 2,
or   NCOL < 2,
or   NT < 1,
or   LDTABL < 6,
or   LDC $<$ NT,
or   TOL < 0.0,
or   IRDF < 0.

IFAIL $= 2$

On entry, IT$(i) < 1$ or IT$(i) >$ NT for some i when NT ≥ 2,
or   no value of IT $= j$ for some $j = 1, 2, \ldots,$ NT, when NT ≥ 2.

IFAIL $= 3$

On entry, the values of Y are constant.

IFAIL $= 4$

A computed standard error is zero due to rounding errors, or the eigenvalue computation failed to converge. Both are unlikely error exits.

IFAIL $= 5$

The treatments are totally confounded with replicates, rows and columns, so the treatment sum of squares and degrees of freedom are zero. The analysis of variance table is not computed, except for replicate, row, column and total sums of squares and degrees of freedom.

IFAIL = 6

>   The residual degrees of freedom or the residual sum of squares are zero, columns 3, 4 and 5 of the analysis of variance table will not be computed and the matrix of standard errors and covariances, C, will not be scaled by s or s^2.

IFAIL = 7

>   The design is disconnected, the standard errors may not be valid. The design may have a nested structure.

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 algorithm used in G04BCF, described in Section 3, achieves greater accuracy than the traditional algorithms based on the subtraction of sums of squares.

8   Parallelism and Performance

G04BCF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

G04BCF 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

To estimate missing values the Healy and Westmacott procedure or its derivatives may be used, see John and Quenouille (1977). This is an iterative procedure in which estimates of the missing values are adjusted by subtracting the corresponding values of the residuals. The new estimates are then used in the analysis of variance. This process is repeated until convergence. A suitable initial value may be the grand mean. When using this procedure IRDF should be set to the number of missing values plus one to obtain the correct degrees of freedom for the residual sum of squares.

For analysis of covariance the residuals are obtained from an analysis of variance of both the response variable and the covariates. The residuals from the response variable are then regressed on the residuals from the covariates using, say, G02CBF or G02DAF. The results from those routines can be used to test for the significance of the covariates. To test the significance of the treatment effects after fitting the covariate, the residual sum of squares from the regression should be compared with the residual sum of squares obtained from the equivalent regression but using the residuals from fitting replicates, rows and columns only.

10   Example

The data for a 5×5 Latin square is input and the ANOVA and treatment means computed and printed. Since the design is orthogonal only one standard error need be printed

10.1 Program Text

```
    Program g04bcfe

!     G04BCF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
    Use nag_library, Only: g04bcf, nag_wp
!     .. Implicit None Statement ..
    Implicit None
!     .. Parameters ..
    Integer, Parameter              :: ldtabl = 6, nin = 5, nout = 6
!     .. Local Scalars ..
    Real (Kind=nag_wp)              :: gmean, tol
    Integer                         :: ifail, irdf, ldc, lit, n, ncol,    &
                                       nrep, nrow, nt
!     .. Local Arrays ..
    Real (Kind=nag_wp), Allocatable :: c(:,:), cmean(:), ef(:), r(:),     &
                                       rmean(:), rpmean(:), tmean(:),     &
                                       wk(:), y(:)
    Real (Kind=nag_wp)              :: tabl(ldtabl,5)
    Integer, Allocatable            :: irep(:), it(:)
!     .. Executable Statements ..
    Write (nout,*) 'G04BCF Example Program Results'
    Write (nout,*)

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

!     Read in the problem size
    Read (nin,*) nrep, nrow, ncol, nt

    n = nrep*nrow*ncol
    If (nt>1) Then
      lit = n
    Else
      lit = 1
    End If
    ldc = nt
    Allocate (irep(nt),y(n),tmean(nt),it(lit),c(ldc,nt),rpmean(nrep), &
      rmean(nrep*nrow),cmean(nrep*ncol),r(n),ef(nt),wk(3*nt))

!     Read in the data
    Read (nin,*) y(1:n)
    If (nt>1) Then
      Read (nin,*) it(1:n)
    End If

!     Use default tolerance
    tol = 0.0E0_nag_wp

!     Use standard degrees of freedom
    irdf = 0

!     Calculate the ANOVA table
    ifail = 0
    Call g04bcf(nrep,nrow,ncol,y,nt,it,gmean,tmean,tabl,ldtabl,c,ldc,irep, &
      rpmean,rmean,cmean,r,ef,tol,irdf,wk,ifail)

!     Display results
    Write (nout,*) ' ANOVA TABLE'
    Write (nout,*)
```

```
      If (nrep>1) Then
        Write (nout,99998) ' Reps        ', tabl(1,1:5)
      End If
      Write (nout,99998) ' Rows         ', tabl(2,1:5)
      Write (nout,99998) ' Columns      ', tabl(3,1:5)
      Write (nout,99998) ' Treatments   ', tabl(4,1:5)
      Write (nout,99998) ' Residual     ', tabl(5,1:3)
      Write (nout,99998) ' Total        ', tabl(6,1:2)
      Write (nout,*)
      Write (nout,*) ' Treatment means'
      Write (nout,*)
      Write (nout,99999) tmean(1:nt)
      Write (nout,*)
      Write (nout,99997) ' S.E. of difference (orthogonal design) = ', c(2,1)

99999 Format (10F10.4)
99998 Format (A,F3.0,2X,3(F10.4,2X),F8.4)
99997 Format (A,F10.4)
      End Program g04bcfe
```

10.2 Program Data

```
G04BCF Example Program Data
1 5 5 5
6.67  7.15  8.29  8.95  9.62
5.40  4.77  5.40  7.54  6.93
7.32  8.53  8.50  9.99  9.68
4.92  5.00  7.29  7.85  7.08
4.88  6.16  7.83  5.38  8.51
5  4  1  3  2
2  5  4  1  3
3  2  5  4  1
1  3  2  5  4
4  1  3  2  5
```

10.3 Program Results

```
G04BCF Example Program Results

 ANOVA TABLE

Rows          4.    29.4231     7.3558     9.0266    0.0013
Columns       4.    22.9950     5.7487     7.0545    0.0037
Treatments    4.     0.5423     0.1356     0.1664    0.9514
Residual     12.     9.7788     0.8149
Total        24.    62.7392

 Treatment means

   7.3180    7.2440    7.2060    6.9000    7.2600

S.E. of difference (orthogonal design) =     0.5709
```