NAG Library Function Document

nag_anova_random (g04bbc)

1 Purpose

nag_anova_random (g04bbc) computes the analysis of variance and treatment means and standard errors for a randomized block or completely randomized design.

2 Specification

```
#include <nag.h>
#include <nagg04.h>
```

```
void nag_anova_random (Integer n, const double y[], Nag_Blocks blocks,
     Integer iblock, Integer nt, const Integer it[], double *gmean,
     double bmean[], double tmean[], double table[], double c[], Integer tdc,
     Integer irep[], double r[], double ef[], double tol, Integer irdf,
     NagError *fail)
```

3 Description

In a completely randomized design the experimental material is divided into a number of units, or plots, to which a treatment can be applied. In a randomized block design the units are grouped into blocks so that the variation within blocks is less than the variation between blocks. If every treatment is applied to one plot in each block it is a complete block design. If there are fewer plots per block than treatments then the design will be an incomplete block design and may be balanced or partially balanced.

For a completely randomized design, with t treatments and n_t plots per treatment, the linear model is

$$y_{ij} = \mu + \tau_j + e_{ij}, \quad j = 1, 2, \ldots, t; \; i = 1, 2, \ldots, n_j,$$

where y_{ij} is the ith observation for the jth treatment, μ is the overall mean, τ_j is the effect of the jth treatment and e_{ij} is the random error term. For a randomized block design, with t treatments and b blocks of k plots, the linear model is

$$y_{ij(l)} = \mu + \beta_i + \tau_l + e_{ij}, \quad i = 1, 2, \ldots, b; \; j = 1, 2, \ldots, k; \; l = 1, 2, \ldots, t,$$

where β_i is the effect of the ith block and the $ij(l)$ notation indicates that the lth treatment is applied to the ith plot in the jth block.

The completely randomized design gives rise to a one-way analysis of variance. The treatments do not have to be equally replicated, i.e., do not have to occur the same number of times. First the overall mean, $\hat{\mu}$, is computed and subtracted from the observations to give $y'_{ij} = y_{ij} - \hat{\mu}$. The estimated treatment effects, $\hat{\tau}_j$ are then computed as the treatment means of the mean adjusted observations, y'_{ij}, and the treatment sum of squares can be computed from the sum of squares of the treatment totals of the y'_{ij} divided by the number of observations per treatment total, n_j. The final residuals are computed as $r_{ij} = y'_{ij} - \hat{\tau}_j$, and, from the residuals, the residual sum of squares is calculated.

For the randomized block design the mean is computed and subtracted from the observations to give $y'_{ij(l)} = y_{ij(l)} - \hat{\mu}$. The estimated block effects, ignoring treatment effects, $\hat{\beta}_i$, are then computed using the block means of the $y'_{ij(l)}$ and the unadjusted sum of squares computed as the sum of squared block totals for the $y'_{ij(l)}$ divided by number of plots per block, k. The block adjusted observations are then computed as $y''_{ij(l)} = y'_{ij(l)} = \hat{\beta}_i$. In the case of the complete block design, with the same replication for each treatment within each block, the blocks and treatments are orthogonal, and so the treatment effects are estimated as the treatment means of the block adjusted observations, $y''_{ij(l)}$. The treatment sum of squares is computed as the sum of squared treatment totals of the $y''_{ij(l)}$ divided by the number of replicates to the treatments, $r = bk/t$. 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 without the same replication for each treatment within each block the treatments and the blocks will not be orthogonal, so the treatments adjusted for blocks need to be computed. The adjusted treatment effects are found as the solution to the equations

$$\left(R - NN^{\mathrm{T}}/k\right)\hat{\tau} = q$$

where q is the vector of the treatment totals for block adjusted observations, $y''_{ij(l)}$, R is a diagonal matrix with R_{ll} equal to the number of times the lth treatment is replicated, and N is the t by b incidence matrix, with N_{lj} equal to the number of times treatment l occurs in block j. The solution to the equations can be written as

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

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

The mean squares are computed as the sum of squares divided by the degrees of freedom. The degrees of freedom for the unadjusted blocks is $b - 1$, for the completely randomized and the complete block 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 the completely randomized or the complete block designs, are given by

$$\mathrm{se}\left(\tau_j - \tau_{j*}\right) = \left(\frac{1}{n_j} + \frac{1}{n_{j*}}\right)s^2$$

where s^2 is the residual mean square and $n_j = n_{j*} = b$ in the complete block design. In the general case the variances of the treatment effects are given by

$$\mathrm{var}\,(\tau) = \Omega s^2$$

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

In the complete block design all the information on the treatment effects is given by the within block analysis. In other designs there may be a loss of information due to the non-orthogonality of treatments and blocks. The efficiency of the within block analysis in these cases is given by the (canonical) efficiency factors, these are the nonzero eigenvalues of the matrix $\left(R - NN^{\mathrm{T}}/k\right)$, 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 some treatments can only be compared using a between block analysis.

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 Arguments

1: **n** – Integer *Input*

On entry: the number of observations, n.

Constraints:

n ≥ 2;
if **iblock** ≥ 2, **n** must be a multiple of **iblock**.

2: **y**[**n**] – const double *Input*

On entry: the observations in the order as described by **blocks** and **nt**.

3: **blocks** – Nag_Blocks *Input*

On entry: **blocks** indicates the block structure.

blocks = Nag_NoBlocks
There are no blocks, i.e., it is a completely randomized design.

blocks = Nag_SerialBlocks
The data should be input by blocks, i.e., **y** must contain the observations for block 1 followed by the observations for block 2, etc.

blocks = Nag_ParallelBlocks
The data is input in parallel with respect to blocks, i.e., **y**[0] must contain the first observation for block 1, **y**[1] must contain the first observation for block 2 ... **y**[**iblock** $-$ 1] must contain the first observation for block **iblock**, **y**[**iblock**] must contain the second observation for block 1, etc.

Constraint: **blocks** = Nag_NoBlocks, Nag_SerialBlocks or Nag_ParallelBlocks.

4: **iblock** – Integer *Input*

On entry: **iblock** indicates the number of blocks, b.

Constraint: if **blocks** = Nag_SerialBlocks or **blocks** = Nag_ParallelBlocks then **iblock** ≥ 2; it is not referenced otherwise.

5: **nt** – Integer *Input*

On entry: the number of treatments, t. If only blocks are required in the analysis then set **nt** $= 1$.

Constraints:

if **iblock** ≥ 2, **nt** ≥ 1;
otherwise **nt** ≥ 2.

6: **it**[**n**] – const Integer *Input*

On entry: **it**$[i-1]$ indicates which of the **nt** treatments plot i received, for $i = 1, 2, \ldots,$ **n**. If **nt** $= 1$, **it** is not referenced.

Constraint: $1 \leq$ **it**$[i-1] \leq$ **nt**, for $i = 1, 2, \ldots,$ **n**.

7: **gmean** – double * *Output*

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

8: **bmean**[**max**(**1**, **iblock**)] – double *Output*

On exit: if **blocks** = Nag_SerialBlocks or Nag_ParallelBlocks, **bmean**$[j-1]$ contains the mean for the jth block, $\hat{\beta}_j$ for $j = 1, 2, \ldots, b$. It is not referenced otherwise.

9: **tmean**[**nt**] – double *Output*

On exit: if $\mathbf{nt} \geq 2$, **tmean**$[l-1]$ 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, $y_{ij(l)} - \hat{\tau}_l$.

10: **table**[**4** × **5**] – double *Output*

Note: the (i,j)th element of the matrix is stored in **table**$[(i-1) \times 5 + j - 1]$.

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 Blocks, row 2 for Treatments, row 3 for Residual and row 4 for Total. Mean squares are computed for all but the Total row; F-statistics and significance are computed for Treatments and Blocks, if present. Any unfilled cells are set to zero.

11: **c**[**nt** × **tdc**] – double *Output*

On exit: if $\mathbf{nt} \geq 2$, 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**$[(i-1) \times \mathbf{tdc} + j - 1]$ 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: **tdc** – Integer *Input*

On entry: the stride separating matrix column elements in the array **c**.

Constraint: $\mathbf{tdc} \geq \mathbf{nt}$.

13: **irep**[**nt**] – Integer *Output*

On exit: if $\mathbf{nt} \geq 2$, the treatment replications, R_{ll}, for $l = 1, 2, \ldots, \mathbf{nt}$.

14: **r**[**n**] – double *Output*

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

15: **ef**[**nt**] – double *Output*

On exit: if $\mathbf{nt} \geq 2$, the canonical efficiency factors.

16: **tol** – double *Input*

On entry: the tolerance value used to check for zero eigenvalues of the matrix Ω. If $\mathbf{tol} = 0.0$ a default value of 10^{-5} is used.

Constraint: $\mathbf{tol} \geq 0.0$.

17: **irdf** – Integer *Input*

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

$\mathbf{irdf} \geq 1$
The degrees of freedom for the total is set to $\mathbf{n} - \mathbf{irdf}$ and the residual degrees of freedom adjusted accordingly.

$\mathbf{irdf} = 0$
The total degrees of freedom for the total is set to $\mathbf{n} - 1$, as usual.

Constraint: $\mathbf{irdf} \geq 0$.

18: **fail** – NagError * *Input/Output*

The NAG error argument (see Section 2.7 in How to Use the NAG Library and its Documentation).

6 Error Indicators and Warnings

NE_2_INT_ARG_LT

On entry, **tdc** $= \langle value \rangle$ while **nt** $= \langle value \rangle$. These arguments must satisfy **tdc** $\geq$ **nt**.

NE_ALLOC_FAIL

Dynamic memory allocation failed.

NE_ARRAY_CONSTANT

On entry, the elements of the array **y** are constant.

NE_BAD_PARAM

On entry, argument **blocks** had an illegal value.

NE_G04BB_CONV

The eigenvalue computation has failed to converge.

This is an unlikely error exit.

NE_G04BB_DESIGN

The design is disconnected; the standard errors may not be valid. The design may be nested.

NE_G04BB_RES_DF

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 the square of s.

NE_G04BB_STDERR

A computed standard error is zero due to rounding errors.

This is an unlikely error exit.

NE_G04BB_TREAT

The treatments are totally confounded with blocks, so the treatment sum of squares and degrees of freedom are zero. The analysis of variance table is not computed, except for block and total sums of squares and degrees of freedom.

NE_INT

On entry, **iblock** $= \langle value \rangle$.
Constraint: **iblock** ≥ 2 when **blocks** = Nag_NoBlocks.

On entry, **nt** $= \langle value \rangle$.
Constraint: **nt** ≥ 2 when **blocks** = Nag_NoBlocks.

NE_INT_2

On entry, **n** $= \langle value \rangle$, **iblock** $= \langle value \rangle$.
Constraint: when **iblock** ≥ 2, **n** must be a multiple of **iblock**.

NE_INT_ARG_LT

On entry, **irdf** $= \langle value \rangle$.
Constraint: **irdf** ≥ 0.

On entry, **n** $= \langle value \rangle$.
Constraint: **n** ≥ 2.

On entry, **nt** $= \langle value \rangle$.
Constraint: **nt** ≥ 1.

NE_INTARR

On entry, **it**$[\langle value \rangle] = \langle value \rangle$.
Constraint: $1 \leq$ **it**$[i-1] \leq$ **nt**, for $i = 1, 2, \ldots,$ **n**.

NE_IT_ARRAY

No value of **it**$[j-1] = j$ for some $j = 1, 2, \ldots,$ **nt**.

NE_REAL_ARG_LT

On entry, **tol** must not be less than 0.0: **tol** $= \langle value \rangle$.

7 Accuracy

The algorithm used by nag_anova_random (g04bbc), described in Section 3, achieves greater accuracy than the traditional algorithms based on the subtraction of sums of squares.

8 Parallelism and Performance

nag_anova_random (g04bbc) is not threaded in any implementation.

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 $\hat{\mu}$. 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 designs such as Latin squares one more of the blocking factors has to be removed in a preliminary analysis before the final analysis using extra calls to nag_anova_random (g04bbc). The residuals from the preliminary analysis are then input to nag_anova_random (g04bbc). In these cases **irdf** should be set to the difference between **n** and the residual degrees of freedom from preliminary analysis. Care should be taken when using this approach as there is no check on the orthogonality of the two analyses.

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, nag_regress_confid_interval (g02cbc) or nag_regsn_mult_linear (g02dac). The results from those functions 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 blocks only.

10 Example

The data, given by John and Quenouille (1977), is for a balanced incomplete block design with 10 blocks and 6 treatments and with 3 plots per block. The observations are the degree of pain experienced and the treatments are penicillin of different potency. The data is input and the analysis of variance table and treatment means are printed.

10.1 Program Text

```
/* nag_anova_random}(g04bbc) Example Program.
 *
 * NAGPRODCODE Version.
 *
 * Copyright 2016 Numerical Algorithms Group.
 *
 * Mark 26, 2016.
 *
 */

#include <nag.h>
#include <nag_stdlib.h>
#include <stdio.h>
#include <nagg04.h>

#define C(I, J)     c[(I) *tdc + J]
#define TABLE(I, J) table[(I) *tdtable + J]
int main(void)
{

  Integer exit_status = 0, i, irdf, j, n, nblock, nt, tdc, tdtable;
  Integer *irep = 0, *it = 0;
  NagError fail;
  Nag_Blocks blocks;
  double gmean, tol;
  double *bmean = 0, *c = 0, *ef = 0, *r = 0, *table = 0, *tmean = 0;
  double *y = 0;

  INIT_FAIL(fail);

  printf("nag_anova_random (g04bbc) Example Program Results\n");

  /* Skip heading in data file */
#ifdef _WIN32
  scanf_s("%*[^\n] ");
#else
  scanf("%*[^\n] ");
#endif
#ifdef _WIN32
  scanf_s("%" NAG_IFMT "%" NAG_IFMT "%" NAG_IFMT "%*[^\n] ", &n, &nt,
          &nblock);
#else
  scanf("%" NAG_IFMT "%" NAG_IFMT "%" NAG_IFMT "%*[^\n] ", &n, &nt, &nblock);
#endif

  if ((nblock >= 2 && !(n % nblock)) || (nblock == 0)) {
    if (nblock != 0) {
      if (!(bmean = NAG_ALLOC(nblock, double)))
      {
        printf("Allocation failure\n");
        exit_status = -1;
        goto END;
      }
      blocks = Nag_SerialBlocks;
    }
    else {
      blocks = Nag_NoBlocks;
    }

    if (!(c = NAG_ALLOC((nt) * (nt), double)) ||
        !(ef = NAG_ALLOC(nt, double)) ||
        !(r = NAG_ALLOC(n, double)) ||
        !(table = NAG_ALLOC((4) * (5), double)) ||
        !(tmean = NAG_ALLOC(nt, double)) ||
        !(y = NAG_ALLOC(n, double)) ||
        !(irep = NAG_ALLOC(nt, Integer)) || !(it = NAG_ALLOC(n, Integer)))
    {
      printf("Allocation failure\n");
      exit_status = -1;
```

```
      goto END;
    }
    tdc = nt;
    tdtable = 5;
  }
  else {
    printf("Invalid nblock or n.\n");
    exit_status = 1;
    return exit_status;
  }
  for (i = 0; i < n; ++i)
#ifdef _WIN32
    scanf_s("%lf", &y[i]);
#else
    scanf("%lf", &y[i]);
#endif
#ifdef _WIN32
  scanf_s("%*[^\n]");
#else
  scanf("%*[^\n]");
#endif
  for (i = 0; i < n; ++i)
#ifdef _WIN32
    scanf_s("%" NAG_IFMT "", &it[i]);
#else
    scanf("%" NAG_IFMT "", &it[i]);
#endif
#ifdef _WIN32
  scanf_s("%*[^\n]");
#else
  scanf("%*[^\n]");
#endif

  tol = 5e-6;
  irdf = 0;

  /* nag_anova_random (g04bbc).
   * General block design or completely randomized design
   */
  nag_anova_random(n, y, blocks, nblock, nt, it, &gmean, bmean, tmean,
                   table, c, tdc, irep, r, ef, tol, irdf, &fail);
  if (fail.code != NE_NOERROR) {
    printf("Error from nag_anova_random (g04bbc).\n%s\n", fail.message);
    exit_status = 1;
    goto END;
  }

  printf("\nANOVA table\n\n");
  printf("   Source        df          SS           MS           F"
         "           Prob\n\n");
  printf(" Blocks      ");
  for (j = 0; j < 5; ++j)
    printf("%10.4f  ", TABLE(0, j));
  printf("\n");

  printf(" Treatments  ");
  for (j = 0; j < 5; ++j)
    printf("%10.4f  ", TABLE(1, j));
  printf("\n");

  printf(" Residual    ");
  for (j = 0; j < 3; ++j)
    printf("%10.4f  ", TABLE(2, j));
  printf("\n");

  printf(" Total       ");
  for (j = 1; j <= 2; ++j)
    printf("%10.4f  ", TABLE(3, j - 1));
  printf("\n");

  printf("\nEfficiency Factors\n\n");
```

```
  for (i = 0; i < nt; ++i)
    printf("%10.5f", ef[i]);
  printf("\n");

  printf("\n%s%10.5f\n", "  Grand Mean", gmean);

  printf("\nTreatment Means\n\n");
  for (i = 1; i <= nt; ++i)
    printf("%10.5f", tmean[i - 1]);
  printf("\n");

  printf("\nStandard errors of differences between means\n\n");
  for (i = 1; i < nt; ++i) {
    for (j = 0; j < i; ++j)
      printf("%10.5f", C(i, j));
    printf("\n");
  }
END:
  NAG_FREE(bmean);
  NAG_FREE(c);
  NAG_FREE(ef);
  NAG_FREE(r);
  NAG_FREE(table);
  NAG_FREE(tmean);
  NAG_FREE(y);
  NAG_FREE(irep);
  NAG_FREE(it);
  return exit_status;
}
```

10.2 Program Data

```
nag_anova_random (g04bbc) Example Program Data
30 6 10                      : n, nt, iblock
1 5 4
5 10 6
2 9 3
4 8 6
2 4 7
6 7 5
5 7 2
7 2 4
8 4 2
10 8 7                       : y
1 2 3
1 2 4
1 3 5
1 4 6
1 5 6
2 3 6
2 4 5
2 5 6
3 4 5
3 4 6                        : it
```

10.3 Program Results

```
nag_anova_random (g04bbc) Example Program Results

ANOVA table

   Source        df         SS          MS          F          Prob

 Blocks          9.0000    60.0000     6.6667     4.7872     0.0039
 Treatments      5.0000   101.7778    20.3556    14.6170     0.0000
 Residual       15.0000    20.8889     1.3926
 Total          29.0000   182.6667

Efficiency Factors
```

```
   0.00000   0.80000   0.80000   0.80000   0.80000   0.80000

  Grand Mean   5.33333

Treatment Means

   2.50000   7.25000   8.08333   5.91667   2.91667   5.33333

Standard errors of differences between means

   0.83444
   0.83444   0.83444
   0.83444   0.83444   0.83444
   0.83444   0.83444   0.83444   0.83444
   0.83444   0.83444   0.83444   0.83444   0.83444
```