NAG Library Function Document

nag_anova_icc (g04gac)

1 Purpose

nag_anova_icc (g04gac) calculates the intraclass correlation (ICC).

2 Specification

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

```
void nag_anova_icc (Nag_ICCModelType mtype, Nag_ICCReliabilityType rtype,
     Integer nrep, Integer nsubj, Integer nrater, const double score[],
     Nag_MissingType mscore, double smiss, double alpha, double *icc,
     double *lci, double *uci, double *fstat, double *df1, double *df2,
     double *pvalue, NagError *fail)
```

3 Description

Many scientific investigations involve assigning a value (score) to a number of objects of interest (subjects). In most instances the method used to score the subject will be affected by measurement error which can affect the analysis and interpretation of the data. When the score is based on the subjective opinion of one or more individuals (raters) the measurement error can be high and therefore it is important to be able to assess its magnitude. One way of doing this is to run a reliability study and calculate the intraclass correlation (ICC).

In a typical reliability study each of a random sample of n_s subjects are scored, independently, by n_r raters. Each rater scores the same subject m times (i.e., there are m replicate scores). The scores, y_{ijk}, for $i = 1, 2, \ldots, n_s$, $j = 1, 2, \ldots, n_r$ and $k = 1, 2, \ldots, m$ can be arranged into m data tables, with the n_s rows of the table, labelled $1, 2, \ldots, n_s$, corresponding to the subjects and the n_r columns of the table, labelled $1, 2, \ldots, n_r$, to the raters. For example the following data, taken from Shrout and Fleiss (1979), shows a typical situation where four raters ($n_r = 4$) have scored six subjects ($n_s = 6$) once, i.e., there has been no replication ($m = 1$).

Subject	Rater			
	1	2	3	4
1	9	2	5	8
2	6	1	3	2
3	8	4	6	8
4	7	1	2	6
5	10	5	6	9
6	6	2	4	7

The term intraclass correlation is a general one and can mean either a measure of interrater reliability, i.e., a measure of how similar the raters are, or intrarater reliability, i.e., a measure of how consistent each rater is.

There are a numerous different versions of the ICC, six of which can be calculated using **nag_anova_icc (g04gac)**. The different versions of the ICC can lead to different conclusions when applied to the same data, it is therefore essential to choose the most appropriate based on the design of the reliability study and whether inter- or intrarater reliability is of interest. The six measures of the ICC are split into three different types of studies, denoted: $\mathrm{ICC}(1,1)$, $\mathrm{ICC}(2,1)$ and $\mathrm{ICC}(3,1)$. This notation ties up with that used by Shrout and Fleiss (1979). Each class of study results in two forms of the ICC, depending on whether inter- or intrarater reliability is of interest.

3.1 ICC$(1,1)$: *One-Factor Design*

The one-factor designs differ, depending on whether inter- or intrarater reliability is of interest:

3.1.1 Interrater reliability

In a one-factor design to measure interrater reliability, each subject is scored by a different set of raters randomly selected from a larger population of raters. Therefore, even though they use the same set of labels each row of the data table is associated with a different set of raters.

A model of the following form is assumed:

$$y_{ijk} = \mu + s_i + \epsilon_{ijk}$$

where s_i is the subject effect and ϵ_{ijk} is the error term, with $s_i \sim N(0, \sigma_s^2)$ and $\epsilon_{ijk} \sim N(0, \sigma_\epsilon^2)$.

The measure of the interrater reliability, ρ, is then given by:

$$\rho = \frac{\hat{\sigma}_s^2}{\hat{\sigma}_s^2 + \hat{\sigma}_\epsilon^2}$$

where $\hat{\sigma}_s$ and $\hat{\sigma}_\epsilon$ are the estimated values of σ_s and σ_ϵ respectively.

3.1.2 Intrarater reliability

In a one-factor design to measure intrarater reliability, each rater scores a different set of subjects. Therefore, even though they use the same set of labels, each column of the data table is associated with a different set of subjects.

A model of the following form is assumed:

$$y_{ijk} = \mu + r_j + \epsilon_{ijk}$$

where r_i is the rater effect and ϵ_{ijk} is the error term, with $r_j \sim N(0, \sigma_r^2)$ and $\epsilon_{ijk} \sim N(0, \sigma_\epsilon^2)$.

The measure of the intrarater reliability, γ, is then given by:

$$\gamma = \frac{\hat{\sigma}_r^2}{\hat{\sigma}_r^2 + \hat{\sigma}_\epsilon^2}$$

where $\hat{\sigma}_r$ and $\hat{\sigma}_\epsilon$ are the estimated values of σ_r and σ_ϵ respectively.

3.2 ICC$(2,1)$: *Random Factorial Design*

In a random factorial design, each subject is scored by the same set of raters. The set of raters have been randomly selected from a larger population of raters.

A model of the following form is assumed:

$$y_{ijk} = \mu + s_i + r_j + (sr)_{ij} + \epsilon_{ijk}$$

where s_i is the subject effect, r_i is the rater effect, $(sr)_{ij}$ is the subject-rater interaction effect and ϵ_{ijk} is the error term, with $s_i \sim N(0, \sigma_s^2)$, $r_j \sim N(0, \sigma_r^2)$, $(sr)_{ij} \sim N(0, \sigma_{sr}^2)$ and $\epsilon_{ijk} \sim N(0, \sigma_\epsilon^2)$.

3.2.1 Interrater reliability

The measure of the interrater reliability, ρ, is given by:

$$\rho = \frac{\hat{\sigma}_s^2}{\hat{\sigma}_s^2 + \hat{\sigma}_r^2 + \hat{\sigma}_{sr}^2 + \hat{\sigma}_\epsilon^2}$$

where $\hat{\sigma}_s$, $\hat{\sigma}_r$, $\hat{\sigma}_{sr}$ and $\hat{\sigma}_\epsilon$ are the estimated values of σ_s, σ_r, σ_{sr} and σ_ϵ respectively.

3.2.2 Intrarater reliability

The measure of the intrarater reliability, γ, is given by:

$$\gamma = \frac{\hat{\sigma}_r^2}{\hat{\sigma}_s^2 + \hat{\sigma}_r^2 + \hat{\sigma}_{sr}^2 + \hat{\sigma}_\epsilon^2}$$

where $\hat{\sigma}_s$, $\hat{\sigma}_r$, $\hat{\sigma}_{sr}$ and $\hat{\sigma}_\epsilon$ are the estimated values of σ_s, σ_r, σ_{sr} and σ_ϵ respectively.

3.3 ICC$(3,1)$: *Mixed Factorial Design*

In a mixed factorial design, each subject is scored by the same set of raters and these are the only raters of interest.

A model of the following form is assumed:

$$y_{ijk} = \mu + s_i + r_j + (sr)_{ij} + \epsilon_{ijk}$$

where s_i is the subject effect, r_i is the fixed rater effect, $(sr)_{ij}$ is the subject-rater interaction effect and ϵ_{ijk} is the error term, with $s_i \sim N(0, \sigma_s^2)$, $\Sigma_{j=1}^{n_r} r_j = 0$, $(sr)_{ij} \sim N(0, \sigma_{sr}^2)$, $\Sigma_{j=1}^{n_r} (sr)_{ij} = 0$ and $\epsilon_{ijk} \sim N(0, \sigma_\epsilon^2)$.

3.3.1 Interrater reliability

The measure of the interrater reliability, ρ, is then given by:

$$\rho = \frac{\hat{\sigma}_s^2 - \hat{\sigma}_{sr}^2/(r-1)}{\hat{\sigma}_s^2 + \hat{\sigma}_{sr}^2 + \hat{\sigma}_\epsilon^2}$$

where $\hat{\sigma}_s$, $\hat{\sigma}_{sr}$ and $\hat{\sigma}_\epsilon$ are the estimated values of σ_s, σ_{sr} and σ_ϵ respectively.

3.3.2 Intrarater reliability

The measure of the intrarater reliability, γ, is then given by:

$$\gamma = \frac{\hat{\sigma}_s^2 + \hat{\sigma}_{sr}^2}{\hat{\sigma}_s^2 + \hat{\sigma}_{sr}^2 + \hat{\sigma}_\epsilon^2}$$

where $\hat{\sigma}_s$, $\hat{\sigma}_{sr}$ and $\hat{\sigma}_\epsilon$ are the estimated values of σ_s, σ_{sr} and σ_ϵ respectively.

As well as an estimate of the ICC, **nag_anova_icc (g04gac)** returns an approximate $(1-\alpha)\%$ confidence interval for the ICC and an F-statistic, f, associated degrees of freedom (ν_1 and ν_2) and p-value, p, for testing that the ICC is zero.

Details on the formula used to calculate the confidence interval, f, ν_1, ν_2, $\hat{\sigma}_s^2$, $\hat{\sigma}_r^2$, $\hat{\sigma}_{sr}^2$ and $\hat{\sigma}_\epsilon^2$ are given in Gwet (2014). In the case where there are no missing data these should tie up with the formula presented in Shrout and Fleiss (1979).

In some circumstances, the formula presented in Gwet (2014) for calculating $\hat{\sigma}_s^2$, $\hat{\sigma}_r^2$, $\hat{\sigma}_{sr}^2$ and $\hat{\sigma}_\epsilon^2$ can result in a negative value being calculated. In such instances, **fail.code** = NW_POTENTIAL_PROBLEM, the offending estimate is set to zero and the calculations continue as normal.

It should be noted that Shrout and Fleiss (1979) also present methods for calculating the ICC based on average scores, denoted ICC$(1,k)$, ICC$(2,k)$ and ICC$(3,k)$. These are not supplied here as multiple replications are allowed ($m > 1$) hence there is no need to average the scores prior to calculating ICC when using **nag_anova_icc (g04gac)**.

4 References

Gwet K L (2014) *Handbook of Inter-rater Reliability* Fourth Edition Advanced Analytics LLC

Shrout P E and Fleiss J L (1979) Intraclass Correlations: Uses in Assessing Rater Reliability *Pyschological Bulletin, Vol 86* **2** 420–428

5 Arguments

1: **mtype** – Nag_ICCModelType *Input*

On entry: indicates which model is to be used.

mtype = Nag_ICC_1
The reliability study is a one-factor design, ICC$(1, 1)$.

mtype = Nag_ICC_2
The reliability study is a random factorial design, ICC$(2, 1)$.

mtype = Nag_ICC_3
The reliability study is a mixed factorial design, ICC$(3, 1)$.

Constraint: **mtype** = Nag_ICC_1, Nag_ICC_2 or Nag_ICC_3.

2: **rtype** – Nag_ICCReliabilityType *Input*

On entry: indicates which type of reliability is required.

rtype = Nag_Inter
Interrater reliability is required.

rtype = Nag_Intra
Intrarater reliability is required.

Constraint: **rtype** = Nag_Inter or Nag_Intra.

3: **nrep** – Integer *Input*

On entry: m, the number of replicates.

Constraints:

if **mtype** = Nag_ICC_2 or Nag_ICC_3 and **rtype** = Nag_Intra, **nrep** ≥ 2;
otherwise **nrep** ≥ 1.

4: **nsubj** – Integer *Input*

On entry: n_s, the number of subjects.

Constraint: **nsubj** ≥ 2.

5: **nrater** – Integer *Input*

On entry: n_r, the number of raters.

Constraint: **nrater** ≥ 2.

6: **score**$[dim]$ – const double *Input*

Note: the dimension, *dim*, of the array **score** must be at least **nrep** $\times$ **nsubj** $\times$ **nrater**.

Where **SCORE**(k, i, j) appears in this document, it refers to the array element **score**$[(j-1) \times$ **nrep** $\times$ **nsubj** $+ (i-1) \times$ **nrep** $+ k - 1]$.

On entry: the matrix of scores, with **SCORE**(k, i, j) being the score given to the ith subject by the jth rater in the kth replicate.

If rater j did not rate subject i at replication k, the corresponding element of **score**, **SCORE**(k, i, j), should be set to **smiss**.

7: **mscore** – Nag_MissingType *Input*

On entry: indicates how missing scores are handled.

mscore = Nag_NoMissing
There are no missing scores.

mscore = Nag_DropMissing
Missing scores in **score** have been set to **smiss**.

Constraint: **mscore** = Nag_NoMissing or Nag_DropMissing.

8: **smiss** – double *Input*

On entry: the value used to indicate a missing score.

If **mscore** = Nag_NoMissing, **smiss** is not referenced and need not be set.

If **mscore** = Nag_DropMissing, care should be taken in the selection of **smiss**, the value used to indicate a missing score. **nag_anova_icc (g04gac)** will treat any score in the inclusive range $\left(1 \pm 0.1^{(\text{nag\_decimal\_digits}-2)}\right) \times$ **smiss** as missing. Alternatively, a NaN (Not A Number) can be used to indicate missing values, in which case the value of **smiss** and any missing values of **score** can be set through a call to **nag_create_nan (x07bbc)**.

9: **alpha** – double *Input*

On entry: α, the significance level used in the construction of the confidence intervals for **icc**.

Constraint: $0 <$ **alpha** < 1.

10: **icc** – double * *Output*

On exit: an estimate of the intraclass correlation to measure either the interrater reliability, ρ, or intrarater reliability, γ, as specified by **mtype** and **rtype**.

11: **lci** – double * *Output*

On exit: an approximate lower limit for the $100(1-\alpha)\%$ confidence interval for the ICC.

12: **uci** – double * *Output*

On exit: an approximate upper limit for the $100(1-\alpha\%)$ confidence interval for the ICC.

In some circumstances it is possible for the estimate of the intraclass correlation to fall outside the region of the approximate confidence intervals. In these cases **nag_anova_icc (g04gac)** returns all calculated values, but raises the warning **fail.code** = NW_POTENTIAL_PROBLEM.

13: **fstat** – double * *Output*

On exit: f, the F-statistic associated with **icc**.

14: **df1** – double * *Output*
15: **df2** – double * *Output*

On exit: ν_1 and ν_2, the degrees of freedom associated with f.

16: **pvalue** – double * *Output*

On exit: $P(F \geq f : \nu_1, \nu_1)$, the upper tail probability from an F distribution.

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

The NAG error argument (see Section 3.7 in How to Use the NAG Library and its Documentation).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 2.3.1.2 in How to Use the NAG Library and its Documentation for further information.

NE_BAD_PARAM

On entry, argument ⟨*value*⟩ had an illegal value.

NE_DEGENERATE

On entry, after adjusting for missing data, **nrater** $= \langle value \rangle$.
Constraint: **nrater** ≥ 2.

On entry, after adjusting for missing data, **nrep** $= \langle value \rangle$.
Constraint: **nrep** ≥ 1.

On entry, after adjusting for missing data, **nrep** $= \langle value \rangle$.
Constraint: when **mtype** = Nag_ICC_2 or Nag_ICC_3 and **rtype** = Nag_Intra, **nrep** ≥ 2.

On entry, after adjusting for missing data, **nsubj** $= \langle value \rangle$.
Constraint: **nsubj** ≥ 2.

Unable to calculate the ICC due to a division by zero.
This is often due to degenerate data, for example all scores being the same.

NE_INT

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

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

On entry, **nrep** $= \langle value \rangle$.
Constraint: when **mtype** = Nag_ICC_2 or Nag_ICC_3 and **rtype** = Nag_Intra, **nrep** ≥ 2.

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

NE_INTERNAL_ERROR

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.
See Section 2.7.6 in How to Use the NAG Library and its Documentation for further information.

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly.
See Section 2.7.5 in How to Use the NAG Library and its Documentation for further information.

NE_REAL

On entry, **alpha** $= \langle value \rangle$.
alpha is too close to either zero or one.
This error is unlikely to occur.

On entry, **alpha** = ⟨*value*⟩.
Constraint: $0 < \textbf{alpha} < 1$.

NW_POTENTIAL_PROBLEM

icc does not fall into the interval [**lci**, **uci**].
All output quantities have been calculated.

On entry, a replicate, subject or rater contained all missing data.
All output quantities have been calculated using the reduced problem size.

The estimate of at least one variance component was negative.
Negative estimates were set to zero and all output quantities calculated as documented.

7 Accuracy

Not applicable.

8 Parallelism and Performance

nag_anova_icc (g04gac) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

nag_anova_icc (g04gac) 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 function. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

None.

10 Example

This example calculates and displays the measure of interrater reliability, ρ, for a one-factor design, ICC$(1,1)$. In addition the 95% confidence interval, F-statistic, degrees of freedom and p-value are presented.

The data is taken from table 2 of Shrout and Fleiss (1979), which has four raters scoring six subjects.

10.1 Program Text

```
/* nag_anova_icc (g04gac) Example Program.
 *
 * Copyright 2017 Numerical Algorithms Group.
 *
 * Mark 26.1, 2017.
 */
/* Pre-processor includes */
#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagg04.h>

int main(void)
{
  /* Integer scalar and array declarations */
  Integer i, j, k, nrater, nsubj, t, nrep;
  Integer exit_status = 0;

  /* Nag Types */
```

```
  NagError fail;
  Nag_ICCModelType mtype;
  Nag_ICCReliabilityType rtype;
  Nag_MissingType mscore;

  /* Double scalar and array declarations */
  double alpha, clevel, df1, df2, fstat, icc, lci, pvalue, uci, smiss;
  double *score = 0;

  /* Character scalar and array declarations */
  char crtype[40], cmtype[40], cmscore[40];

  /* Initialize the error structure */
  INIT_FAIL(fail);

  printf("nag_anova_icc (g04gac) Example Program Results\n\n");

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

  /* Read in the problem size */
#ifdef _WIN32
  scanf_s("%39s%39s%" NAG_IFMT "%" NAG_IFMT "%" NAG_IFMT "%*[^\n] ", cmtype,
          (unsigned)_countof(cmtype), crtype, (unsigned)_countof(crtype),
          &nrep, &nsubj, &nrater);
#else
  scanf("%39s%39s%" NAG_IFMT "%" NAG_IFMT "%" NAG_IFMT "%*[^\n] ", cmtype,
        crtype, &nrep, &nsubj, &nrater);
#endif
  mtype = (Nag_ICCModelType) nag_enum_name_to_value(cmtype);
  rtype = (Nag_ICCReliabilityType) nag_enum_name_to_value(crtype);

  /* Read in the values used to identify missing scores */
#ifdef _WIN32
  scanf_s("%39s%lf%*[^\n] ", cmscore, (unsigned)_countof(cmscore), &smiss);
#else
  scanf("%39s%lf%*[^\n] ", cmscore, &smiss);
#endif
  mscore = (Nag_MissingType) nag_enum_name_to_value(cmscore);

  /* Allocate memory */
  if (!(score = NAG_ALLOC(nsubj*nrater*nrep, double)))
  {
    printf("Allocation failure\n");
    exit_status = -1;
    goto END;
  }

  /* Read in the rating data */
  for (k = 0; k < nrep; k++) {
    for (i = 0; i < nsubj; i++) {
      for (j = 0; j < nrater; j++) {
        t = j * nrep * nsubj + i * nrep + k;
#ifdef _WIN32
        scanf_s("%lf", &score[t]);
#else
        scanf("%lf", &score[t]);
#endif
      }
#ifdef _WIN32
      scanf_s("%*[^\n] ");
#else
      scanf("%*[^\n] ");
#endif
    }
  }
```

```
  /* Read in alpha for the confidence interval */
#ifdef _WIN32
  scanf_s("%lf%*[^\n] ", &alpha);
#else
  scanf("%lf%*[^\n] ", &alpha);
#endif

  /* Call nag_anova_icc (g04gac) to calculate the intraclass correlation */
  nag_anova_icc(mtype,rtype,nrep,nsubj,nrater,score,mscore,smiss,alpha,&icc,
                &lci,&uci,&fstat,&df1,&df2,&pvalue,&fail);
  if (fail.code != NE_NOERROR) {
    printf("Error from nag_anova_icc (g04gac).\n%s\n", fail.message);
    exit_status = -1;
    if (fail.code != NW_POTENTIAL_PROBLEM)
      goto END;
  }

  /* Display the results */
  printf("Intraclass Correlation           : %5.2f\n", icc);
  clevel = 100.0*(1.0-alpha);
  printf("Lower Limit for %4.1f%% CI         : %5.2f\n", clevel, lci);
  printf("Upper Limit for %4.1f%% CI         : %5.2f\n", clevel, uci);
  printf("F statistic                      : %5.2f\n", fstat);
  printf("Degrees of Freedom 1             : %5.1f\n", df1);
  printf("Degrees of Freedom 2             : %5.1f\n", df2);
  printf("p-value                          : %5.3f\n", pvalue);

END:
  NAG_FREE(score);

  return (exit_status);
}
```

10.2 Program Data

```
nag_anova_icc (g04gac) Example Program Data
Nag_ICC_1 Nag_Inter 1 6 4 :: mtype,rtype,nrep,nsubj,nrater
Nag_NoMissing -99         :: mscore,smiss
 9  2  5  8
 6  1  3  2
 8  4  6  8
 7  1  2  6
10  5  6  9
 6  2  4  7               :: end of score
0.05                      :: alpha
```

10.3 Program Results

```
nag_anova_icc (g04gac) Example Program Results

Intraclass Correlation           :  0.17
Lower Limit for 95.0% CI         : -0.13
Upper Limit for 95.0% CI         :  0.72
F statistic                      :  1.79
Degrees of Freedom 1             :   5.0
Degrees of Freedom 2             :  18.0
p-value                          : 0.165
```
