

# NAG C Library Function Document

## nag\_friedman\_test (g08aec)

### 1 Purpose

nag\_friedman\_test (g08aec) performs the Friedman two-way analysis of variance by ranks on  $k$  related samples of size  $n$ .

### 2 Specification

```
#include <nag.h>
#include <nagg08.h>

void nag_friedman_test (Integer k, Integer n, const double x[], Integer tdx,
                        double *fr, double *p, NagError *fail)
```

### 3 Description

The Friedman test investigates the score differences between  $k$  matched samples of size  $n$ , the scores in the  $i$ th sample being denoted by:

$$x_{i1}, x_{i2}, \dots, x_{in}.$$

(Thus the sample scores may be regarded as a two-way table with  $k$  rows and  $n$  columns.) The hypothesis under test,  $H_0$ , often called the null hypothesis, is that the samples come from the same population, and this is to be tested against the alternative hypothesis  $H_1$  that they come from different populations.

The test is based on the observed distribution of score rankings between the matched observations in different samples.

The test proceeds as follows:

- The scores in each column are ranked,  $r_{ij}$  denoting the rank within column  $j$  of the observation in row  $i$ . Average ranks are assigned to tied scores.
- The ranks are summed over each row to give rank sums  $t_i = \sum_{j=1}^k r_{ij}$ , for  $i = 1, 2, \dots, k$ .
- The Friedman test statistic  $FR$  is computed, where

$$FR = \frac{12}{nk(k+1)} \sum_{i=1}^k \left\{ t_i - \frac{1}{2}n(k+1) \right\}^2.$$

nag\_friedman\_test returns the value of  $FR$ , and also an approximation,  $p$ , to the significance of this value. ( $FR$  approximately follows a  $\chi_{k-1}^2$  distribution, so large values of  $FR$  imply rejection of  $H_0$ ).  $H_0$  is rejected by a test of chosen size  $\alpha$  if  $p < \alpha$ . The approximation  $p$  is acceptable unless  $k = 4$  and  $n < 5$ , or  $k = 3$  and  $n < 10$ , or  $k = 2$  and  $n < 20$ ; for  $k = 3$  or  $4$ , tables should be consulted (e.g., **n** of Siegel (1956)); for  $k = 2$  the Sign test (see nag\_sign\_test (g08aac)) or Wilcoxon test (see nag\_wilcoxon\_test (g08agc)) is in any case more appropriate.

### 4 Parameters

- 1: **k** – Integer *Input*  
*On entry:* the number of samples,  $k$ .  
*Constraint:* **k** > 1.

- 2: **n** – Integer *Input*  
*On entry:* the size of each sample,  $n$ .  
*Constraint:*  $n \geq 1$ .
- 3: **x[k][tdx]** – const double *Input*  
*On entry:*  $x[i-1][j-1]$  must be set to the value,  $x_{ij}$ , of observation  $j$  in sample  $i$ , for  $i = 1, 2, \dots, k$ ;  $j = 1, 2, \dots, n$ .
- 4: **tdx** – Integer *Input*  
*On entry:* the first dimension of the array **x** as declared in the function from which `nag_friedman_test` is called.  
*Constraint:*  $tdx \geq n$ .
- 5: **fr** – double \* *Output*  
*On exit:* the value of the Friedman test statistic,  $FR$ .
- 6: **p** – double \* *Output*  
*On exit:* the approximate significance,  $p$ , of the Friedman test statistic.
- 7: **fail** – NagError \* *Input/Output*  
The NAG error parameter (see the Essential Introduction).

## 5 Error Indicators and Warnings

### NE\_INT\_ARG\_LT

On entry, **n** must not be less than 1: **n** = *<value>*.

### NE\_INT\_ARG\_LE

On entry, **k** must not be less than or equal to 1: **k** = *<value>*.

### NE\_2\_INT\_ARG\_LT

On entry, **tdx** = *<value>* while **n** = *<value>*.  
These parameters must satisfy  $tdx \geq n$ .

### NE\_ALLOC\_FAIL

Memory allocation failed.

### 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 consult NAG for assistance.

## 6 Further Comments

The time taken by the routine is approximately proportional to the product  $nk$ .

If  $k = 2$ , the Sign test (see `nag_sign_test` (g08aac)) or Wilcoxon test (see `nag_wilcoxon_test` (g08agc)) is more appropriate.

### 6.1 Accuracy

For estimates of the accuracy of the significance  $p$ , see `nag_prob_chi_sq` (g01ecc). The  $\chi^2$  approximation is acceptable unless  $k = 4$  and  $n < 5$ , or  $k = 3$  and  $n < 10$ , or  $k = 2$  and  $n < 20$ .

## 6.2 References

Siegel S (1956) *Non-parametric Statistics for the Behavioral Sciences* McGraw-Hill

## 7 See Also

nag\_prob\_chi\_sq (g01ecc)  
 nag\_sign\_test (g08aac)  
 nag\_wilcoxon\_test (g08agc)

## 8 Example

This example is taken from page 169 of Siegel (1956). The data relate to training scores of three matched samples of 18 rats, trained under three different patterns of reinforcement.

### 8.1 Program Text

```

/* nag_friedman_test (g08aec) Example Program.
 *
 * Copyright 2000 Numerical Algorithms Group.
 *
 * Mark 6, 2000.
 */

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagg08.h>

int main (void)
{
  double fr, sig, *x=0;
  Integer i, ix, j, k, n;
  Integer exit_status=0;
  NagError fail;

#define X(I,J) x[((I)-1)*n +(J)-1]

  INIT_FAIL(fail);
  Vprintf("g08aec Example Program Results\n");

  /* Skip heading in data file */
  Vscanf("%*[\n]");

  n = 18;
  k = 3;
  ix = k;
  if (!(x = NAG_ALLOC(ix*n, double)))
  {
    Vprintf("Allocation failure\n");
    exit_status = -1;
    goto END;
  }
  for (i = 1; i <= ix; ++i)
    for (j = 1; j <= n; ++j)
      Vscanf("%lf", &X(i,j));

  Vprintf("\nFriedman test\n");

```

```

Vprintf("\nData values\n");
Vprintf("\n  Group  Group  Group\n");
Vprintf("    1      2      3\n");
for (j = 1; j <= 18; ++j)
  {
    for (i = 1; i <= 3; ++i)
Vprintf("%7.1f", X(i,j));
    Vprintf("\n");
  }
g08aec(k, n, x, n, &fr, &sig, &fail);
if (fail.code != NE_NOERROR)
  {
    Vprintf("Error from g08aec.\n%s\n", fail.message);
    exit_status = 1;
    goto END;
  }
Vprintf("\n");
Vprintf("%s%6.3f\n", "Test statistic      ", fr);
Vprintf("%s%6ld\n", "Degrees of freedom    ", k-1);
Vprintf("%s%6.3f\n", "Significance          ", sig);
END:
if (x) NAG_FREE(x);
return exit_status;
}

```

## 8.2 Program Data

g08aec Example Program Data

```

1 2 1 1 3 2 3 1 3 3 2 2 3 2 2.5 3 3 2
3 3 3 2 1 3 2 3 1 1 3 3 2 3 2.5 2 2 3
2 1 2 3 2 1 1 2 2 2 1 1 1 1 1 1 1 1

```

## 8.3 Program Results

g08aec Example Program Results

Friedman test

Data values

Group	Group	Group
1	2	3
1.0	3.0	2.0
2.0	3.0	1.0
1.0	3.0	2.0
1.0	2.0	3.0
3.0	1.0	2.0
2.0	3.0	1.0
3.0	2.0	1.0
1.0	3.0	2.0
3.0	1.0	2.0
3.0	1.0	2.0
2.0	3.0	1.0
2.0	3.0	1.0
3.0	2.0	1.0
2.0	3.0	1.0
2.5	2.5	1.0
3.0	2.0	1.0
3.0	2.0	1.0

2.0    3.0    1.0

Test statistic	8.583
Degrees of freedom	2
Significance	0.014

---