NAG Library Function Document

nag_ranks_and_scores (g01dhc)

1 Purpose

nag_ranks_and_scores (g01dhc) computes the ranks, Normal scores, an approximation to the Normal scores or the exponential scores as requested by you.

2 Specification

```c
#include <nag.h>
#include <nagg01.h>

void nag_ranks_and_scores (Nag_Scores scores, Nag_Ties ties, Integer n,
const double x[], double r[], NagError *fail)
```

3 Description

nag_ranks_and_scores (g01dhc) computes one of the following scores for a sample of observations, \(x_1, x_2, \ldots, x_n\).

1. Rank Scores
   
   The ranks are assigned to the data in ascending order, that is the \(i\)th observation has score \(s_i = k\) if it is the \(k\)th smallest observation in the sample.

2. Normal Scores
   
   The Normal scores are the expected values of the Normal order statistics from a sample of size \(n\). If \(x_i\) is the \(k\)th smallest observation in the sample, then the score for that observation, \(s_i\), is \(E(Z_k)\) where \(Z_k\) is the \(k\)th order statistic in a sample of size \(n\) from a standard Normal distribution and \(E\) is the expectation operator.

3. Blom, Tukey and van der Waerden Scores

   These scores are approximations to the Normal scores. The scores are obtained by evaluating the inverse cumulative Normal distribution function, \(\Phi^{-1}(\cdot)\), at the values of the ranks scaled into the interval (0, 1) using different scaling transformations.

   The Blom scores use the scaling transformation \(\frac{r_i - \frac{3}{8}}{n + \frac{1}{4}}\) for the rank \(r_i\), for \(i = 1, 2, \ldots, n\). Thus the Blom score corresponding to the observation \(x_i\) is
   
   \[
   s_i = \Phi^{-1}\left(\frac{r_i - \frac{3}{8}}{n + \frac{1}{4}}\right).
   \]

   The Tukey scores use the scaling transformation \(\frac{r_i - \frac{1}{2}}{n + \frac{1}{2}}\); the Tukey score corresponding to the observation \(x_i\) is
   
   \[
   s_i = \Phi^{-1}\left(\frac{r_i - \frac{1}{2}}{n + \frac{1}{2}}\right).
   \]

   The van der Waerden scores use the scaling transformation \(\frac{r_i}{n + 1}\); the van der Waerden score corresponding to the observation \(x_i\) is
   
   \[
   s_i = \Phi^{-1}\left(\frac{r_i}{n + 1}\right).
   \]

   The van der Waerden scores may be used to carry out the van der Waerden test for testing for differences between several population distributions, see Conover (1980).
4. Savage Scores

The Savage scores are the expected values of the exponential order statistics from a sample of size $n$. They may be used in a test discussed by Savage (1956) and Lehmann (1975). If $x_i$ is the $k$th smallest observation in the sample, then the score for that observation is

$$s_i = E(Y_k) = \frac{1}{n} + \frac{1}{n-1} + \cdots + \frac{1}{n-k+1},$$

where $Y_k$ is the $k$th order statistic in a sample of size $n$ from a standard exponential distribution and $E$ is the expectation operator.

Ties may be handled in one of five ways. Let $x_{i(j)}$, for $i = 1, 2, \ldots, m$, denote $m$ tied observations, that is $x_{i(1)} = x_{i(2)} = \cdots = x_{i(m)}$ with $t(1) < t(2) < \cdots < t(m)$. If the rank of $x_{i(1)}$ is $k$, then if ties are ignored the rank of $x_{i(j)}$ will be $k + j - 1$. Let the scores ignoring ties be $s_{i(1)}^*, s_{i(2)}^*, \ldots, s_{i(m)}^*$. Then the scores, $s_{i(j)}$, for $i = 1, 2, \ldots, m$, may be calculated as follows:

- if averages are used, then $s_{i(j)} = \frac{1}{m} \sum_{j=1}^{m} s_{i(j)}^* / m$;
- if the lowest score is used, then $s_{i(j)} = s_{i(1)}^*$;
- if the highest score is used, then $s_{i(j)} = s_{i(m)}^*$;
- if ties are to be broken randomly, then $s_{i(j)} = s_{i(I)}^*$, where $I \in \{\text{random permutation of } 1, 2, \ldots, m\}$;
- if ties are to be ignored, then $s_{i(j)} = s_{i(j)}^*$.

4 References

Blom G (1958) Statistical Estimates and Transformed Beta-variables Wiley
Lehmann E L (1975) Nonparametrics: Statistical Methods Based on Ranks Holden–Day

5 Arguments

1: scores – Nag_Scores

On entry: indicates which of the following scores are required.

scores = Nag_RankScores
The ranks.
scores = Nag_NormalScores
The Normal scores, that is the expected value of the Normal order statistics.
scores = Nag_BlomScores
The Blom version of the Normal scores.
scores = Nag_TukeyScores
The Tukey version of the Normal scores.
scores = Nag_WaerdenScores
The van der Waerden version of the Normal scores.
The Savage scores, that is the expected value of the exponential order statistics.

Constraint: \( \text{scores} = \text{Nag\_RankScores}, \text{Nag\_NormalScores}, \text{Nag\_BlomScores}, \text{Nag\_TukeyScores}, \text{Nag\_WaerdenScores} \) or \( \text{Nag\_SavageScores} \).

2: \( \text{ties} \) – Nag\_Ties

On entry: indicates which of the following methods is to be used to assign scores to tied observations.

- \( \text{ties} = \text{Nag\_AverageTies} \)
  The average of the scores for tied observations is used.
- \( \text{ties} = \text{Nag\_LowestTies} \)
  The lowest score in the group of ties is used.
- \( \text{ties} = \text{Nag\_HighestTies} \)
  The highest score in the group of ties is used.
- \( \text{ties} = \text{Nag\_RandomTies} \)
  The repeatable random number generator is used to randomly untie any group of tied observations.
- \( \text{ties} = \text{Nag\_IgnoreTies} \)
  Any ties are ignored, that is the scores are assigned to tied observations in the order that they appear in the data.

Constraint: \( \text{ties} = \text{Nag\_AverageTies}, \text{Nag\_LowestTies}, \text{Nag\_HighestTies}, \text{Nag\_RandomTies} \) or \( \text{Nag\_IgnoreTies} \).

3: \( n \) – Integer

On entry: \( n \), the number of observations.

Constraint: \( n \geq 1 \).

4: \( x[n] \) – const double

On entry: the sample of observations, \( x_i \), for \( i = 1, 2, \ldots, n \).

5: \( r[n] \) – double

On exit: contains the scores, \( s_i \), for \( i = 1, 2, \ldots, n \), as specified by \( \text{scores} \).

6: \( \text{fail} \) – NagError *

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

**NE\_ALLOC\_FAIL**

Dynamic memory allocation failed.

See Section 3.2.1.2 in the Essential Introduction for further information.

**NE\_BAD\_PARAM**

On entry, argument \( \langle \text{value} \rangle \) had an illegal value.

**NE\_INT\_ARG\_LT**

On entry, \( n = \langle \text{value} \rangle \).

Constraint: \( n \geq 1 \).
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.

An unexpected error has been triggered by this function. Please contact NAG.
See Section 3.6.6 in the Essential Introduction for further information.

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly.
See Section 3.6.5 in the Essential Introduction for further information.

7 Accuracy

For scores = Nag_RankScores, the results should be accurate to machine precision.
For scores = Nag_SavageScores, the results should be accurate to a small multiple of machine precision.
For scores = Nag_NormalScores, the results should have a relative accuracy of at least max(100 × ε, 10⁻³) where ε is the machine precision.
For scores = Nag_BlomScores, Nag_TukeyScores or Nag_WaerdenScores, the results should have a relative accuracy of at least max(10 × ε, 10⁻¹²).

8 Parallelism and Performance

nag_ranks_and_scores (g01dhc) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

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

If more accurate Normal scores are required nag_normal_scores_exact (g01dac) should be used with appropriate settings for the input argument etol.

10 Example

This example computes and prints the Savage scores for a sample of five observations. The average of the scores of any tied observations is used.

10.1 Program Text

/* nag_ranks_and_scores (g01dhc) Example Program. *
* Copyright 2014 Numerical Algorithms Group. *
* Mark 4, 1996.
* Mark 8 revised, 2004. *
*/
#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nagg01.h>

int main(void)
{
    Integer exit_status = 0, i, n;

NagError fail;
double *r = 0, *x = 0;
INIT_FAIL(fail);
printf("nag_ranks_and_scores (g01dhc) Example Program Results\n\n");
/* Skip heading in data file */
#ifdef _WIN32
    scanf_s("%*[\n ] ");
#else
    scanf("%*[\n ] ");
#endif
#ifdef _WIN32
    scanf("%"NAG_IFMT" , &n);
#else
    scanf("%"NAG_IFMT" , &n);
#endif
if (n >= 1)
{
    if (!(r = NAG_ALLOC(n, double)) ||
        !(x = NAG_ALLOC(n, double)))
    {
        printf("Allocation failure\n");
        exit_status = -1;
        goto END;
    }
}
else
{
    printf("Invalid n.\n");
    exit_status = 1;
    return exit_status;
}
for (i = 1; i <= n; ++i)
    #ifdef _WIN32
        scanf_s("%lf " , &x[i - 1]);
    #else
        scanf("%lf " , &x[i - 1]);
    #endif
/* nag_ranks_and_scores (g01dhc).
* Ranks, Normal scores, approximate Normal scores or
* exponential (Savage) scores
*/
nag_ranks_and_scores(Nag_SavageScores, Nag_AverageTies, n, x, r,
    &fail);
if (fail.code != NE_NOERROR)
{
    printf("Error from nag_ranks_and_scores (g01dhc).
    fail.message");
    exit_status = 1;
    goto END;
}
printf("The Savage Scores : \n");
printf(" (Average scores are used for tied observations)\n\n");
for (i = 1; i <= n; ++i)
    printf("%10.4f\n", r[i - 1]);
END:
NAG_FREE(r);
NAG_FREE(x);
return exit_status;
}
10.2 Program Data

nag_ranks_and_scores (g01dhc) Example Program Data
5
2 0 2 2 0

10.3 Program Results

nag_ranks_and_scores (g01dhc) Example Program Results

The Savage Scores :
(Average scores are used for tied observations)

1.4500
0.3250
1.4500
1.4500
0.3250