NAG C Library Function Document

nag_normal_scores_exact (g01dac)

1 Purpose

nag_normal_scores_exact (g01dac) computes a set of Normal scores, i.e., the expected values of an ordered set of independent observations from a Normal distribution with mean 0.0 and standard deviation 1.0.

2 Specification

```
void nag_normal_scores_exact (Integer n, double pp[], double etol, double *errest,
     NagError *fail)
```

3 Description

If a sample of n observations from any distribution (which may be denoted by $x_1, x_2, \ldots, x_n$), is sorted into ascending order, the rth smallest value in the sample is often referred to as the rth '**order statistic**', sometimes denoted by $x_{(r)}$ (see Kendall and Stuart (1969)).

The order statistics therefore have the property

$$x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(n)}.$$

(If $n = 2r + 1$, x_{r+1} is the sample median.)

For samples originating from a known distribution, the distribution of each order statistic in a sample of given size may be determined. In particular, the expected values of the order statistics may be found by integration. If the sample arises from a Normal distribution, the expected values of the order statistics are referred to as the '**Normal scores**'. The Normal scores provide a set of reference values against which the order statistics of an actual data sample of the same size may be compared, to provide an indication of Normality for the sample. A plot of the data against the scores gives a normal probability plot. Normal scores have other applications; for instance, they are sometimes used as alternatives to ranks in nonparametric testing procedures.

nag_normal_scores_exact (g01dac) computes the rth Normal score for a given sample size n as

$$E(x_{(r)}) = \int_{-\infty}^{\infty} x_r dG_r,$$

where

$$dG_r = \frac{A_r^{r-1}(1 - A_r)^{n-r} dA_r}{\beta(r, n - r + 1)}, \quad A_r = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x_r} e^{-t^2/2}\, dt, \quad r = 1, 2, \ldots, n,$$

and β denotes the complete Beta function.

The function attempts to evaluate the scores so that the estimated error in each score is less than the value **etol** specified by the user. All integrations are performed in parallel and arranged so as to give good speed and reasonable accuracy.

4 References

Kendall M G and Stuart A (1969) *The Advanced Theory of Statistics (Volume 1)* (3rd Edition) Griffin

5 Parameters

1: **n** – Integer *Input*

On entry: the size of the set, n.

Constraint: **n** > 0.

2: **pp**[**n**] – double *Output*

On exit: the Normal scores. **pp**$[i-1]$ contains the value $E(x_{(i)})$, for $i = 1, 2, \ldots, n$.

3: **etol** – double *Input*

On entry: the maximum value for the estimated absolute error in the computed scores.

Constraint: **etol** > 0.0.

4: **errest** – double * *Output*

On exit: a computed estimate of the maximum error in the computed scores (see Section 7).

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

The NAG error parameter (see the Essential Introduction).

6 Error Indicators and Warnings

NE_INT

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

NE_ERROR_ESTIMATE

The function was unable to estimate the scores with estimated error less than **etol**.

NE_REAL

On entry, **etol** $= \langle value \rangle$.
Constraint: **etol** > 0.0.

NE_ALLOC_FAIL

Memory allocation failed.

NE_BAD_PARAM

On entry, parameter $\langle value \rangle$ had an illegal value.

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.

7 Accuracy

Errors are introduced by evaluation of the functions dG_r and errors in the numerical integration process. Errors are also introduced by the approximation of the true infinite range of integration by a finite range $[a, b]$ but a and b are chosen so that this effect is of lower order than that of the other two factors. In order to estimate the maximum error the functions dG_r are also integrated over the range $[a, b]$. nag_normal_scores_exact (g01dac) returns the estimated maximum error as

$$\textbf{errest} = \max_r \left[\max(|a|, |b|) \times \left| \int_a^b dG_r - 1.0 \right| \right].$$

8 Further Comments

The time taken by nag_normal_scores_exact (g01dac) depends on **etol** and **n**. For a given value of **etol** the timing varies approximately linearly with **n**.

9 Example

The program below generates the Normal scores for samples of size 5, 10, 15, and prints the scores and the computed error estimates.

9.1 Program Text

```
/* nag_normal_scores_exact (g01dac) Example Program.
 *
 * Copyright 2001 Numerical Algorithms Group.
 *
 * Mark 7, 2001.
 */

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagg01.h>

int main(void)
{

/* Scalars */
  double errest, etol;
  Integer exit_status, i, j, n, nmax;
  NagError fail;

/* Arrays */
  double *pp=0;

  Vprintf("g01dac Example Program Results\n");
  INIT_FAIL(fail);
  exit_status = 0;
  etol = 0.001;
  nmax = 15;

  /* Allocate memory */
  if ( !(pp = NAG_ALLOC(nmax, double))  )
    {
      Vprintf("Allocation failure\n");
      exit_status = -1;
      goto END;
    }

  for (j = 5; j <= nmax; j += 5)
    {
      n = j;
      g01dac(n, pp, etol, &errest, &fail);

      Vprintf("\nSet size = %2ld\n\n", n);
      Vprintf("Error tolerance (input) = %13.3e\n\n", etol);
      Vprintf("Error estimate (output) = %13.3e\n\n", errest);
      Vprintf("Normal scores\n");
      for (i = 1; i <= n; ++i)
        {
          Vprintf("%10.3f", pp[i - 1]);
          Vprintf(i%5 == 0 || i == n ?"\n":" ");
```

```
        }
      Vprintf("\n");
      if (fail.code != NE_NOERROR)
        {
          Vprintf("Error from g01dac.\n%s\n", fail.message);
          exit_status = 1;
        }
    }
 END:
  if (pp) NAG_FREE(pp);

  return exit_status;
}
```

9.2 Program Data

None.

9.3 Program Results

```
g01dac Example Program Results

Set size =  5

Error tolerance (input) =     1.000e-03

Error estimate (output) =     9.080e-09

Normal scores
    -1.163     -0.495      0.000      0.495      1.163

Set size = 10

Error tolerance (input) =     1.000e-03

Error estimate (output) =     1.484e-08

Normal scores
    -1.539     -1.001     -0.656     -0.376     -0.123
     0.123      0.376      0.656      1.001      1.539

Set size = 15

Error tolerance (input) =     1.000e-03

Error estimate (output) =     2.218e-08

Normal scores
    -1.736     -1.248     -0.948     -0.715     -0.516
    -0.335     -0.165      0.000      0.165      0.335
     0.516      0.715      0.948      1.248      1.736
```