NAG C Library Function Document

nag_rngs_students_t (g05lbc)

1 Purpose

nag_rngs_students_t (g05lbc) generates a vector of pseudo-random numbers taken from a Student's t-distribution with ν degrees of freedom.

2 Specification

```
void nag_rngs_students_t (Integer df, Integer n, double x[], Integer igen,
     Integer iseed[], NagError *fail)
```

3 Description

The distribution has PDF (probability density function)

$$f(x) = \frac{\left(\frac{\nu-1}{2}\right)!}{\left(\frac{1}{2}\nu - 1\right)!\sqrt{\pi\nu}\left(1 + \frac{x^2}{\nu}\right)^{\frac{1}{2}(\nu+1)}}.$$

nag_rngs_students_t (g05lbc) calculates the values

$$y_i\sqrt{\frac{\nu}{z_i}}, \quad i = 1, \dots, n$$

where the y_i are generated by nag_rngs_normal (g05lac) from a Normal distribution with mean 0 and variance 1.0, and the z_i are generated by nag_rngs_gamma (g05lfc) from a gamma distribution with parameters $\frac{1}{2}\nu$ and 2 (i.e., from a χ^2 distribution with ν degrees of freedom).

One of the initialisation functions nag_rngs_init_repeatable (g05kbc) (for a repeatable sequence if computed sequentially) or nag_rngs_init_nonrepeatable (g05kcc) (for a non-repeatable sequence) must be called prior to the first call to nag_rngs_students_t (g05lbc).

4 References

Knuth D E (1981) *The Art of Computer Programming (Volume 2)* (2nd Edition) Addison–Wesley

5 Parameters

1: **df** – Integer *Input*

On entry: the number of degrees of freedom, ν, of the distribution.

Constraint: **df** ≥ 1.

2: **n** – Integer *Input*

On entry: the number, n, of pseudo-random numbers to be generated.

Constraint: **n** ≥ 0.

3: **x**[dim] – double *Output*

Note: the dimension, dim, of the array **x** must be at least max$(1, \mathbf{n})$.

On exit: the n pseudo-random numbers from the specified Student's t-distribution.

4: **igen** – Integer *Input*

On entry: must contain the identification number for the generator to be used to return a pseudo-random number and should remain unchanged following initialisation by a prior call to one of the functions nag_rngs_init_repeatable (g05kbc) or nag_rngs_init_nonrepeatable (g05kcc).

5: **iseed**[4] – Integer *Input/Output*

On entry: contains values which define the current state of the selected generator.

On exit: contains updated values defining the new state of the selected generator.

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

The NAG error parameter (see the Essential Introduction).

6 Error Indicators and Warnings

NE_INT

On entry, $\mathbf{n} = \langle value \rangle$.
Constraint: $\mathbf{n} \geq 0$.

On entry, $\mathbf{df} = \langle value \rangle$.
Constraint: $\mathbf{df} \geq 1$.

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

Not applicable.

8 Further Comments

The time taken by nag_rngs_students_t (g05lbc) increases with ν.

9 Example

The example program prints 5 pseudo-random numbers from a Student's t-distribution with five degrees of freedom, generated by a single call to nag_rngs_students_t (g05lbc), after initialisation by nag_rngs_init_repeatable (g05kbc).

9.1 Program Text

```
/* nag_rngs_students_t(g05lbc) Example Program.
 *
 * Copyright 2001 Numerical Algorithms Group.
 *
 * Mark 7, 2001.
 */

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagg05.h>

int main(void)
```

```
{
  /* Scalars */
  Integer i, igen, n;
  Integer exit_status=0;
  NagError fail;

  /* Arrays */
  double *x=0;
  Integer  iseed[4];

  INIT_FAIL(fail);
  Vprintf("g05lbc Example Program Results\n\n");

  n = 5;
  /* Allocate memory */
  if ( !(x = NAG_ALLOC(n, double)) )
    {
      Vprintf("Allocation failure\n");
      exit_status = -1;
      goto END;
    }

  /* Initialise the seed to a repeatable sequence */
  iseed[0] = 1762543;
  iseed[1] = 9324783;
  iseed[2] = 42344;
  iseed[3] = 742355;

  /* igen identifies the stream. */
  igen = 1;
  g05kbc(&igen, iseed);
  g05lbc(5, n, x, igen, iseed, &fail);
  if (fail.code != NE_NOERROR)
    {
      Vprintf("Error from g05lbc.\n%s\n", fail.message);
      exit_status = 1;
      goto END;
    }
  for (i = 0; i < n; ++i)
    {
      Vprintf("%10.4f\n", x[i]);
    }

 END:
  if (x) NAG_FREE(x);
  return exit_status;
}
```

9.2 Program Data

None.

9.3 Program Results

```
g05lbc Example Program Results

    2.3726
   -0.8473
   -0.0452
   -1.3595
   -0.5932
```