NAG Library Function Document

nag_bivariate_students_t (g01hcc)

1 Purpose

nag_bivariate_students_t (g01hcc) returns probabilities for the bivariate Student's t-distribution.

2 Specification

```
#include <nag.h>
#include <nagg01.h>
double nag_bivariate_students_t (Nag_TailProbability tail, const double a[],
       const double b[], Integer df, double rho, NagError *fail)
```

3 Description

Let the vector random variable $X = (X_1, X_2)^{\mathrm{T}}$ follow a bivariate Student's t-distribution with degrees of freedom ν and correlation ρ, then the probability density function is given by

$$f\left(X : \nu, \rho\right) = \frac{1}{2\pi\sqrt{1-\rho^2}}\left(1 + \frac{X_1^2 + X_2^2 - 2\rho X_1 X_2}{\nu(1-\rho^2)}\right)^{-\nu/2-1}.$$

The lower tail probability is defined by:

$$P\left(X_1 \leq b_1, X_2 \leq b_2 : \nu, \rho\right) = \int_{-\infty}^{b_1}\int_{-\infty}^{b_2} f\left(X : \nu, \rho\right) dX_2 dX_1.$$

The upper tail probability is defined by:

$$P\left(X_1 \geq a_1, X_2 \geq a_2 : \nu, \rho\right) = \int_{a_1}^{\infty}\int_{a_2}^{\infty} f\left(X : \nu, \rho\right) dX_2 dX_1.$$

The central probability is defined by:

$$P\left(a_1 \leq X_1 \leq b_1, a_2 \leq X_2 \leq b_2 : \nu, \rho\right) = \int_{a_1}^{b_1}\int_{a_2}^{b_2} f\left(X : \nu, \rho\right) dX_2 dX_1.$$

Calculations use the Dunnet and Sobel (1954) method, as described by Genz (2004).

4 References

Dunnet C W and Sobel M (1954) A bivariate generalization of Student's t-distribution, with tables for certain special cases *Biometrika* **41** 153–169

Genz A (2004) Numerical computation of rectangular bivariate and trivariate Normal and t probabilities *Statistics and Computing* **14** 151–160

5 Arguments

1: **tail** – Nag_TailProbability *Input*

On entry: indicates which probability is to be returned.

tail = Nag_LowerTail
The lower tail probability is returned.

tail = Nag_UpperTail
The upper tail probability is returned.

tail = Nag_Central
The central probability is returned.

Constraint: **tail** = Nag_LowerTail, Nag_UpperTail or Nag_Central.

2: **a**[**2**] – const double *Input*

On entry: if **tail** = Nag_Central or Nag_UpperTail, the lower bounds a_1 and a_2.

If **tail** = Nag_LowerTail, **a** is not referenced.

3: **b**[**2**] – const double *Input*

On entry: if **tail** = Nag_Central or Nag_LowerTail, the upper bounds b_1 and b_2.

If **tail** = Nag_UpperTail, **b** is not referenced.

Constraint: if **tail** = Nag_Central, $a_i < b_i$, for $i = 1, 2$.

4: **df** – Integer *Input*

On entry: ν, the degrees of freedom of the bivariate Student's t-distribution.

Constraint: **df** ≥ 1.

5: **rho** – double *Input*

On entry: ρ, the correlation of the bivariate Student's t-distribution.

Constraint: $-1.0 \leq$ **rho** ≤ 1.0.

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

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

6 Error Indicators and Warnings

NE_BAD_PARAM

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

NE_INT

On entry, **df** = ⟨*value*⟩.
Constraint: **df** ≥ 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.

NE_REAL

On entry, **rho** = ⟨*value*⟩.
Constraint: $-1.0 \leq$ **rho** ≤ 1.0.

NE_REAL_2

On entry, **b**$[i-1] \leq$ **a**$[i-1]$ for central probability, for some $i = 1, 2$.

7 Accuracy

Accuracy of the algorithm implemented here is discussed in comparison with algorithms based on a generalized Placket formula by Genz (2004), who recommends the Dunnet and Sobel method. This implementation should give a maximum absolute error of the order of 10^{-16}.

8 Parallelism and Performance

Not applicable.

9 Further Comments

None.

10 Example

This example calculates the bivariate Student's t probability given the choice of tail and degrees of freedom, correlation and bounds.

10.1 Program Text

```
/* nag_bivariate_students_t (g01hcc) Example Program.
 *
 * Mark 23 Release. NAG Copyright 2011.
 */
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagg01.h>

int main(void)
{
  /* Scalars */
  Integer             df, exit_status = 0, ierr;
  double              prob, rho;
  /* Arrays */
  char                nag_enum_arg[14];
  double              a[2], b[2];
  /* NAG types */
  Nag_TailProbability tail;
  NagError           fail;

  printf("%s\n\n",
          "nag_bivariate_students_t (g01hcc) Example Program Results");

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

  /* Display headers */
  printf("%-8s%2s%-8s%2s%-8s%2s%-8s%2s%-4s%2s%-8s%2s%-14s%2s%-8s\n\n",
          "a1", " ", "b1", " ", "a2", " ", "b2", " ", "df", " ", "rho", " ",
          "Tail", " ",  "p");

  while (1)
  {
    ierr = scanf("%14s", nag_enum_arg);
    if (ierr == EOF || ierr < 1)
    {
        break;
    }

    /* Initialize limits */
    a[0] = a[1] = b[0] = b[1] = 0.0;

    /* nag_enum_name_to_value (x04nac).
     * Converts NAG enum member name to value
     */
    tail = (Nag_TailProbability)nag_enum_name_to_value(nag_enum_arg);

    /* Read parameter values*/
    switch (tail)
```

```
    {
    case Nag_LowerTail:
        scanf("%ld%lf%lf%lf", &df, &rho, b, b+1);
        break;
    case Nag_Central:
        scanf("%ld%lf%lf%lf%lf%lf", &df, &rho, a, b, a+1, b+1);
        break;
    case Nag_UpperTail:
        scanf("%ld%lf%lf%lf", &df, &rho, a, a+1);
        break;
    default:
        printf(" %s\n", "Invalid tail specification in data file");
        exit_status = -1;
        goto END;
    }
    scanf("%*[^\n]");

    /* Calculate probablity for the bivariate Student's t-distribution */
    INIT_FAIL(fail);
    /* nag_bivariate_students_t (g01hcc) */
    prob = nag_bivariate_students_t(tail, a, b, df, rho, &fail);

    /* Display results */
    switch (tail)
    {
    case Nag_LowerTail:
        printf("%-8s%2s%-8g%2s%-8s%2s%-8g",
                "-Inf", " ", b[0], " ", "-Inf", " ", b[1]);
        break;
    case Nag_Central:
        printf("%-8g%2s%-8g%2s%-8g%2s%-8g",
                a[0], " ", b[0], " ", a[1], " ", b[1]);
        break;
    case Nag_UpperTail:
        printf("%-8g%2s%-8s%2s%-8g%2s%-8s",
                a[0], " ", "Inf", " ", a[1], " ", "Inf");
        break;
    default:
      {
        printf("Invalid tail specification.\n");
        exit_status = -1;
        goto END;
      }
    }

    printf("%2s%-4ld%2s%-8g%2s%-14s%2s%-8.4f\n",
            " ", df, " ", rho, " ", nag_enum_arg, " ", prob);
  }

  END:
  return exit_status;
}
```

10.2 Program Data

```
nag_bivariate_students_t (g01hcc) Example Program Data
Nag_LowerTail  8  0.6   4.0 0.8         : tail df rho      b[i], i=0,1
Nag_Central   12 -0.2 -40.0 2.0 0.0 4.0 : tail df rho (a, b)[i], i=0,1
Nag_UpperTail  2  0.3  -2.0 8.0         : tail df rho      a[i], i=0,1
```

10.3 Program Results

```
nag_bivariate_students_t (g01hcc) Example Program Results

a1        b1        a2        b2        df    rho       Tail            p

-Inf      4         -Inf      0.8       8     0.6       Nag_LowerTail   0.7764
-40       2         0         4         12    -0.2      Nag_Central     0.4876
-2        Inf       8         Inf       2     0.3       Nag_UpperTail   0.0059
```