nag_prob_students_t (g01ebc)

1. Purpose

nag_prob_students_t (g01ebc) returns the lower tail, upper tail or two-tail probability for the Student's t-distribution with real degrees of freedom.

2. Specification

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

double nag_prob_students_t(Nag_TailProbability tail, double t, double df,
       NagError *fail)
```

3. Description

The lower tail probability for the Student's t-distribution with ν degrees of freedom, $P(T \leq t : \nu)$, is defined by

$$P(T \leq t : \nu) = \frac{\Gamma((\nu+1)/2)}{\sqrt{\pi\nu}\Gamma(\nu/2)} \int_{-\infty}^{t} \left[1 + \frac{T^2}{\nu}\right]^{-(\nu+1)/2} dT, \quad \nu \geq 1.$$

Computationally, there are two situations:

(a) when $\nu < 20$, a transformation of the beta distribution, $P_\beta(B \leq \beta : a, b)$ is used;

$$P(T \leq t : \nu) = \tfrac{1}{2} P_\beta \left(B \leq \frac{\nu}{\nu + t^2} : \nu/2, \tfrac{1}{2} \right) \quad \text{when } t < 0.0$$

or

$$P(T \leq t : \nu) = \tfrac{1}{2} + \tfrac{1}{2} P_\beta \left(B \geq \frac{\nu}{\nu + t^2} : \nu/2, \tfrac{1}{2} \right) \quad \text{when } t > 0.0$$

(b) when $\nu \geq 20$, an asymptotic normalising expansion of the Cornish–Fisher type is used to evaluate the probability, see Hill (1970).

4. Parameters

tail

Input: indicates which tail the returned probability should represent.
If **tail** = **Nag_UpperTail**, the upper tail probability is returned, i.e., $P(T \geq t : \nu)$.
If **tail** = **Nag_LowerTail**, the lower tail probability is returned, i.e., $P(T \leq t : \nu)$.
If **tail** = **Nag_TwoTailSignif**, the two tail (significance level) probability is returned, i.e., $P(T \geq |t| : \nu) + P(T \leq -|t| : \nu)$.
If **tail** = **Nag_TwoTailConfid**, the two tail (confidence interval) probability is returned, i.e., $P(T \leq |t| : \nu) - P(T \leq -|t| : \nu)$.
Constraint: **tail** = **Nag_UpperTail** or **Nag_LowerTail** or **Nag_TwoTailSignif** or **Nag_TwoTailConfid**.

t

Input: the value of the Student's t variate, t.

df

Input: the degrees of freedom, ν, of the Student's t-distribution.
Constraint: **df** ≥ 1.

fail

The NAG error parameter, see the Essential Introduction to the NAG C Library.

5. Error Indications and Warnings

On any of the error conditions listed below nag_prob_students_t returns 0.0.

NE_BAD_PARAM
On entry, parameter **tail** had an illegal value.

NE_REAL_ARG_LT
On entry, **df** must not be less than 1.0: **df** = ⟨*value*⟩.

6. Further Comments

The probabilities could also be obtained by using the appropriate transformation to a Beta distribution (see Abramowitz and Stegun, 1965) and using nag_prob_beta_dist (g01eec). This function allows the user to set the required accuracy.

6.1. Accuracy

The computed probability should to be accurate to 5 significant places for reasonable probabilities but there will be some loss of accuracy for very low probabilities (less than 10^{-10}), see Hill (1970).

6.2. References

Abramowitz M and Stegun I A (1965) *Handbook of Mathematical Functions* Dover Publications, New York ch 26.
Hastings N A J and Peacock J B (1975) *Statistical Distributions* Butterworth.
Hill G W (1970) Student's *t*-distribution *Commun. ACM* **13** (10) 617–619.

7. See Also

None.

8. Example

Values from, and degrees of freedom for Student's *t*-distributions are read along with the required tail. The probabilities are calculated and printed until the end of data is reached.

8.1. Program Text

```
/* nag_prob_students_t(g01ebc) Example Program
 *
 * Copyright 1996 Numerical Algorithms Group.
 *
 * Mark 4, 1996.
 */

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

main()
{
  double df, prob, t;
  int i;
  static Nag_TailProbability tail[4] = {Nag_LowerTail, Nag_UpperTail,
                                        Nag_TwoTailSignif, Nag_TwoTailConfid};
  static char *tailmess[] = { "Nag_LowerTail", "Nag_UpperTail",
                              "Nag_TwoTailSignif", "Nag_TwoTailConfid"};

  Vprintf("g01ebc Example Program Results\n\n");
  /*  Skip heading in data file */
  Vscanf("%*[^\n]");
  Vprintf("    t      df     prob      tail\n\n");
  while (scanf("%lf %lf %ld\n", &t, &df, &i) != EOF)
    {
      prob = g01ebc(tail[i], t, df, NAGERR_DEFAULT);
      Vprintf(" %6.3f%8.3f%8.4f  %s\n", t, df, prob, tailmess[i]);
    }
  exit(EXIT_SUCCESS);
}
```

8.2. Program Data

```
g01ebc Example Program Data
0.85  20.0  0
0.85  20.0  2
0.85  20.0  3
0.85  20.0  1
```

8.3. Program Results

```
g01ebc Example Program Results

    t      df     prob      tail

  0.850  20.000  0.7973  Nag_LowerTail
  0.850  20.000  0.4054  Nag_TwoTailSignif
  0.850  20.000  0.5946  Nag_TwoTailConfid
  0.850  20.000  0.2027  Nag_UpperTail
```