

NAG Library Function Document

nag_prob_non_central_students_t (g01gbc)

1 Purpose

nag_prob_non_central_students_t (g01gbc) returns the lower tail probability for the noncentral Student's t -distribution.

2 Specification

```
#include <nag.h>
#include <nagg01.h>
double nag_prob_non_central_students_t (double t, double df, double delta,
    double tol, Integer max_iter, NagError *fail)
```

3 Description

The lower tail probability of the noncentral Student's t -distribution with ν degrees of freedom and noncentrality parameter δ , $P(T \leq t : \nu; \delta)$, is defined by

$$P(T \leq t : \nu; \delta) = C_\nu \int_0^\infty \left(\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\alpha u - \delta} e^{-x^2/2} dx \right) u^{\nu-1} e^{-u^2/2} du, \quad \nu > 0.0$$

with

$$C_\nu = \frac{1}{\Gamma(\frac{1}{2}\nu) 2^{(\nu-2)/2}}, \quad \alpha = \frac{t}{\sqrt{\nu}}.$$

The probability is computed in one of two ways.

(i) When $t = 0.0$, the relationship to the normal is used:

$$P(T \leq t : \nu; \delta) = \frac{1}{\sqrt{2\pi} \int_\delta} \int_\delta^\infty e^{-u^2/2} du.$$

(ii) Otherwise the series expansion described in Equation 9 of Amos (1964) is used. This involves the sums of confluent hypergeometric functions, the terms of which are computed using recurrence relationships.

4 References

Amos D E (1964) Representations of the central and non-central t -distributions *Biometrika* **51** 451–458

5 Arguments

- | | | |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| 1: | t – double
<i>On entry:</i> t , the deviate from the Student's t -distribution with ν degrees of freedom. | <i>Input</i> |
| 2: | df – double
<i>On entry:</i> ν , the degrees of freedom of the Student's t -distribution.
<i>Constraint:</i> df \geq 1.0. | <i>Input</i> |
| 3: | delta – double
<i>On entry:</i> δ , the noncentrality argument of the Students t -distribution. | <i>Input</i> |

- 4: **tol** – double *Input*
On entry: the absolute accuracy required by you in the results. If `nag_prob_non_central_students_t` (g01gbc) is entered with **tol** greater than or equal to 1.0 or less than $10 \times \textit{machine precision}$ (see `nag_machine_precision` (X02AJC)), then the value of $10 \times \textit{machine precision}$ is used instead.
- 5: **max_iter** – Integer *Input*
On entry: the maximum number of terms that are used in each of the summations.
Suggested value: 100. See Section 9 for further comments.
Constraint: **max_iter** \geq 1.
- 6: **fail** – NagError * *Input/Output*
The NAG error argument (see Section 2.7 in How to Use the NAG Library and its Documentation).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.

See Section 3.2.1.2 in How to Use the NAG Library and its Documentation for further information.

NE_INT_ARG_LT

On entry, **max_iter** = $\langle \textit{value} \rangle$.

Constraint: **max_iter** \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 How to Use the NAG Library and its Documentation 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 How to Use the NAG Library and its Documentation for further information.

NE_PROB_LIMIT

The probability is too close to 0 or 1.

NE_PROBABILITY

The probability is too small to calculate accurately.

NE_REAL_ARG_LT

On entry, **df** = $\langle \textit{value} \rangle$.

Constraint: **df** \geq 1.0.

NE_SERIES

One of the series has failed to converge with **max_iter** = $\langle \textit{value} \rangle$ and **tol** = $\langle \textit{value} \rangle$. Reconsider the requested tolerance and/or the maximum number of iterations.

7 Accuracy

The series described in Amos (1964) are summed until an estimated upper bound on the contribution of future terms to the probability is less than `tol`. There may also be some loss of accuracy due to calculation of gamma functions.

8 Parallelism and Performance

`nag_prob_non_central_students_t` (g01gbc) is not threaded in any implementation.

9 Further Comments

The rate of convergence of the series depends, in part, on the quantity $t^2/(t^2 + \nu)$. The smaller this quantity the faster the convergence. Thus for large t and small ν the convergence may be slow. If ν is an integer then one of the series to be summed is of finite length.

If two tail probabilities are required then the relationship of the t -distribution to the F -distribution can be used:

$$F = T^2, \lambda = \delta^2, \nu_1 = 1 \quad \text{and} \quad \nu_2 = \nu,$$

and a call made to `nag_prob_non_central_f_dist` (g01gdc).

Note that `nag_prob_non_central_students_t` (g01gbc) only allows degrees of freedom greater than or equal to 1 although values between 0 and 1 are theoretically possible.

10 Example

This example reads values from, and degrees of freedom for, and noncentrality arguments of the noncentral Student's t -distributions, calculates the lower tail probabilities and prints all these values until the end of data is reached.

10.1 Program Text

```
/* nag_prob_non_central_students_t (g01gbc) Example Program.
 *
 * NAGPRODCODE Version.
 *
 * Copyright 2016 Numerical Algorithms Group.
 *
 * Mark 26, 2016.
 */

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

int main(void)
{
    Integer exit_status = 0, max_iter;
    NagError fail;
    double delta, df, prob, t, tol;

    INIT_FAIL(fail);

    printf("nag_prob_non_central_students_t (g01gbc) Example Program Results"
           "\n\n");

    /* Skip heading in data file */
#ifdef _WIN32
    scanf_s("%*[\n]");
#else
    scanf("%*[\n]");
#endif
}
```

```

printf("      t      df      delta      prob\n\n");
tol = 5e-6;
max_iter = 50;
#ifdef _WIN32
while ((scanf_s("%lf %lf %lf %*[\n]", &t, &df, &delta)) != EOF)
#else
while ((scanf("%lf %lf %lf %*[\n]", &t, &df, &delta)) != EOF)
#endif
{
/* nag_prob_non_central_students_t (g01gbc).
* Computes probabilities for the non-central Student's
* t-distribution
*/
prob = nag_prob_non_central_students_t(t, df, delta, tol, max_iter,
&fail);

if (fail.code == NE_NOERROR)
printf(" %8.3f%8.3f%8.3f%8.4f\n", t, df, delta, prob);
else {
printf("Error from nag_prob_non_central_students_t (g01gbc).\n%s\n",
fail.message);
exit_status = 1;
goto END;
}
}
END:
return exit_status;
}

```

10.2 Program Data

```

nag_prob_non_central_students_t (g01gbc) Example Program Data
-1.528  20.0  2.0           :t df delta
-0.188   7.5  1.0           :t df delta
 1.138  45.0  0.0           :t df delta

```

10.3 Program Results

```

nag_prob_non_central_students_t (g01gbc) Example Program Results

```

t	df	delta	prob
-1.528	20.000	2.000	0.0003
-0.188	7.500	1.000	0.1189
1.138	45.000	0.000	0.8694
