NAG Library Function Document

nag_prob_chi_sq (g01ecc)

1 Purpose

nag_prob_chi_sq (g01ecc) returns the lower or upper tail probability for the χ^2-distribution with real degrees of freedom.

2 Specification

```
#include <nag.h>
#include <nagg01.h>
double nag_prob_chi_sq (Nag_TailProbability tail, double x, double df,
       NagError *fail)
```

3 Description

The lower tail probability for the χ^2-distribution with ν degrees of freedom, $P(X \leq x : \nu)$ is defined by:

$$P(X \leq x : \nu) = \frac{1}{2^{\nu/2}\Gamma(\nu/2)} \int_{0.0}^{x} X^{\nu/2-1} e^{-X/2}\, dX, \quad x \geq 0, \nu > 0.$$

To calculate $P(X \leq x : \nu)$ a transformation of a gamma distribution is employed, i.e., a χ^2-distribution with ν degrees of freedom is equal to a gamma distribution with scale parameter 2 and shape parameter $\nu/2$.

4 References

Abramowitz M and Stegun I A (1972) *Handbook of Mathematical Functions* (3rd Edition) Dover Publications

Hastings N A J and Peacock J B (1975) *Statistical Distributions* Butterworth

5 Arguments

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

On entry: indicates whether the upper or lower tail probability is required.

tail = Nag_LowerTail
The lower tail probability is returned, i.e., $P(X \leq x : \nu)$.

tail = Nag_UpperTail
The upper tail probability is returned, i.e., $P(X \geq x : \nu)$.

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

2: **x** – double *Input*

On entry: x, the value of the χ^2 variate with ν degrees of freedom.

Constraint: **x** ≥ 0.0.

3: **df** – double *Input*

On entry: ν, the degrees of freedom of the χ^2-distribution.

Constraint: **df** > 0.0.

4: **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_ALG_NOT_CONV

The series used to calculate the gamma probabilities has failed to converge. The result returned should represent an approximation to the solution.

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_BAD_PARAM

On entry, argument ⟨*value*⟩ 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 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_REAL_ARG_LE

On entry, **df** = ⟨*value*⟩.
Constraint: **df** > 0.0.

NE_REAL_ARG_LT

On entry, **x** = ⟨*value*⟩.
Constraint: **x** ≥ 0.0.

7 Accuracy

A relative accuracy of five significant figures is obtained in most cases.

8 Parallelism and Performance

nag_prob_chi_sq (g01ecc) is not threaded in any implementation.

9 Further Comments

For higher accuracy the transformation described in Section 3 may be used with a direct call to nag_incomplete_gamma (s14bac).

10 Example

Values from various χ^2-distributions are read, the lower tail probabilities calculated, and all these values printed out, until the end of data is reached.

10.1 Program Text

```
/* nag_prob_chi_sq (g01ecc) Example Program.
 *
 * NAGPRODCODE Version.
 *
 * Copyright 2016 Numerical Algorithms Group.
 *
 * Mark 26, 2016.
 */

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

int main(void)
{
  Integer exit_status = 0;
  double df, prob, x;
  NagError fail;

  INIT_FAIL(fail);

  /* Skip heading in data file */
#ifdef _WIN32
  scanf_s("%*[^\n]");
#else
  scanf("%*[^\n]");
#endif
  printf("nag_prob_chi_sq (g01ecc) Example Program Results\n");
  printf("  x      df     prob\n\n");
#ifdef _WIN32
  while (scanf_s("%lf %lf", &x, &df) != EOF)
#else
  while (scanf("%lf %lf", &x, &df) != EOF)
#endif
  {
    /* nag_prob_chi_sq (g01ecc).
     * Probabilities for chi^2 distribution
     */
    prob = nag_prob_chi_sq(Nag_LowerTail, x, df, &fail);
    if (fail.code != NE_NOERROR) {
      printf("Error from nag_prob_chi_sq (g01ecc).\n%s\n", fail.message);
      exit_status = 1;
      goto END;
    }
    printf("%6.3f%8.3f%8.4f\n", x, df, prob);
  }

END:
  return exit_status;
}
```

10.2 Program Data

```
nag_prob_chi_sq (g01ecc) Example Program Data
  8.26   20.0
  6.2     7.5
 55.76   45.0
```

10.3 Program Results

```
nag_prob_chi_sq (g01ecc) Example Program Results
  x      df     prob

 8.260  20.000  0.0100
 6.200   7.500  0.4279
55.760  45.000  0.8694
```