NAG Library Function Document

nag_prob_f_dist (g01edc)

1 Purpose

nag_prob_f_dist (g01edc) returns the probability for the lower or upper tail of the F or variance-ratio distribution with real degrees of freedom.

2 Specification

```
#include <nag.h>
#include <nagg01.h>
double nag_prob_f_dist (Nag_TailProbability tail, double f, double df1,
       double df2, NagError *fail)
```

3 Description

The lower tail probability for the F, or variance-ratio distribution, with ν_1 and ν_2 degrees of freedom, $P(F \leq f : \nu_1, \nu_2)$, is defined by:

$$P(F \leq f : \nu_1, \nu_2) = \frac{\nu_1^{\nu_1/2}\nu_2^{\nu_2/2}\Gamma((\nu_1+\nu_2)/2)}{\Gamma(\nu_1/2)\Gamma(\nu_2/2)} \int_0^f F^{(\nu_1-2)/2}(\nu_1 F + \nu_2)^{-(\nu_1+\nu_2)/2} dF,$$

for $\nu_1,\ \nu_2 > 0,\ f \geq 0$.

The probability is computed by means of a transformation to a beta distribution, $P_\beta(B \leq \beta : a, b)$:

$$P(F \leq f : \nu_1, \nu_2) = P_\beta\left(B \leq \frac{\nu_1 f}{\nu_1 f + \nu_2} : \nu_1/2, \nu_2/2\right)$$

and using a call to nag_prob_beta_dist (g01eec).

For very large values of both ν_1 and ν_2, greater than 10^5, a normal approximation is used. If only one of ν_1 or ν_2 is greater than 10^5 then a χ^2 approximation is used, see Abramowitz and Stegun (1972).

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 an upper or lower tail probability is required.

tail = Nag_LowerTail
The lower tail probability is returned, i.e., $P(F \leq f : \nu_1, \nu_2)$.

tail = Nag_UpperTail
The upper tail probability is returned, i.e., $P(F \geq f : \nu_1, \nu_2)$.

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

2: **f** – double *Input*

On entry: f, the value of the F variate.

Constraint: **f** ≥ 0.0.

3: **df1** – double *Input*

On entry: the degrees of freedom of the numerator variance, ν_1.

Constraint: **df1** > 0.0.

4: **df2** – double *Input*

On entry: the degrees of freedom of the denominator variance, ν_2.

Constraint: **df2** > 0.0.

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

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

6 Error Indicators and Warnings

On any of the error conditions listed below except NE_PROBAB_CLOSE_TO_TAIL nag_prob_f_dist (g01edc) returns 0.0.

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 3.2.1.2 in the Essential Introduction 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 the Essential Introduction 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 the Essential Introduction for further information.

NE_PROBAB_CLOSE_TO_TAIL

The probability is too close to 0.0 or 1.0. **f** is too far out into the tails for the probability to be evaluated exactly. The result tends to approach 1.0 if f is large, or 0.0 if f is small. The result returned is a good approximation to the required solution.

NE_REAL_ARG_LE

On entry, **df1** = ⟨*value*⟩ and **df2** = ⟨*value*⟩.
Constraint: **df1** > 0.0 and **df2** > 0.0.

NE_REAL_ARG_LT

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

7 Accuracy

The result should be accurate to five significant digits.

8 Parallelism and Performance

Not applicable.

9 Further Comments

For higher accuracy nag_prob_beta_dist (g01eec) can be used along with the transformations given in Section 3.

10 Example

This example reads values from, and degrees of freedom for, a number of F-distributions and computes the associated lower tail probabilities.

10.1 Program Text

```
/* nag_prob_f_dist (g01edc) Example Program.
 *
 * Copyright 2014 Numerical Algorithms Group.
 *
 * Mark 1, 1990.
 */

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

int main(void)
{
  Integer  exit_status = 0;
  double   df1, df2, f, prob;
  NagError fail;

  INIT_FAIL(fail);

  /* Skip heading in data file */
#ifdef _WIN32
  scanf_s("%*[^\n]");
#else
  scanf("%*[^\n]");
#endif
  printf("nag_prob_f_dist (g01edc) Example Program Results\n");
  printf("  f      df1      df2     prob\n\n");
#ifdef _WIN32
  while (scanf_s("%lf %lf %lf", &f, &df1, &df2) != EOF)
#else
  while (scanf("%lf %lf %lf", &f, &df1, &df2) != EOF)
#endif
    {
      /* nag_prob_f_dist (g01edc).
       * Probabilities for F-distribution
       */
      prob = nag_prob_f_dist(Nag_LowerTail, f, df1, df2, &fail);
      if (fail.code != NE_NOERROR)
        {
          printf("Error from nag_prob_f_dist (g01edc).\n%s\n",
                  fail.message);
          exit_status = 1;
          goto END;
        }
```

```
      printf("%6.3f%8.3f%8.3f%8.4f\n", f, df1, df2, prob);
    }

 END:
  return exit_status;
}
```

10.2 Program Data

```
nag_prob_f_dist (g01edc) Example Program Data
  5.5   1.5   25.5
 39.9   1.0    1.0
 2.5   20.25   1.0
```

10.3 Program Results

```
nag_prob_f_dist (g01edc) Example Program Results
  f       df1      df2     prob

 5.500   1.500  25.500  0.9837
39.900   1.000   1.000  0.9000
 2.500  20.250   1.000  0.5342
```