

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 (1965).

4. Parameters**tail**

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

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

If **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**.

f

Input: the value of the F variate, f .

Constraint: **f** ≥ 0.0 .

df1

Input: the degrees of freedom of the numerator variance, ν_1 .

Constraint: **df1** > 0.0 .

df2

Input: the degrees of freedom of the denominator variance, ν_2 .

Constraint: **df2** > 0.0 .

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 except **NE_PROBAB_CLOSE_TO_TAIL** nag_prob_f_dist returns 0.0.

NE_BAD_PARAM

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

NE_REAL_ARG_LT

On entry, **f** must not be less than 0.0: **f** = $\langle value \rangle$.

NE_REAL_ARG_LE

On entry, **df1** must not be less than or equal to 0.0: **df1** = $\langle value \rangle$.

On entry, **df2** must not be less than or equal to 0.0: **df2** = $\langle value \rangle$.

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.

6. Further Comments

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

6.1. Accuracy

The result should be accurate to 5 significant digits.

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.

7. See Also

nag_prob_beta_dist (g01eec)

8. Example

Values from, and degrees of freedom for F -distributions are read, the lower-tail probabilities computed, and all these values printed, until the end of data is reached.

8.1. Program Text

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

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

main()
{
    double df1, df2, f, prob;
    static NagError fail;

    /* Skip heading in data file */
    Vscanf("%*[^\\n]");
    Vprintf("g01edc Example Program Results\\n");
    Vprintf("  f      df1      df2      prob\\n\\n");
```

```

while (scanf("%lf %lf %lf", &f, &df1, &df2) != EOF)
{
    prob = g01edc(Nag_LowerTail, f, df1, df2, &fail);

    if (fail.code==NE_NOERROR)
        Vprintf("%6.3f%8.3f%8.3f%8.4f\n", f, df1, df2, prob);
    else
        Vprintf("%6.3f%8.3f%8.3f%8.4f\n Note: %s\n",f,df1,df2,prob,
                fail.message);
    }
    exit(EXIT_SUCCESS);
}

```

8.2. Program Data

```

g01edc Example Program Data
  5.5   1.5  25.5
 39.9   1.0   1.0
  2.5  20.25  1.0

```

8.3. Program Results

```

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

```
