

NAG Library Function Document

nag_prob_1_sample_ks (g01eyc)

1 Purpose

nag_prob_1_sample_ks (g01eyc) returns the upper tail probability associated with the one sample Kolmogorov–Smirnov distribution.

2 Specification

```
#include <nag.h>
#include <nagg01.h>
double nag_prob_1_sample_ks (Integer n, double d, NagError *fail)
```

3 Description

Let $S_n(x)$ be the sample cumulative distribution function and $F_0(x)$ the hypothesised theoretical distribution function.

nag_prob_1_sample_ks (g01eyc) returns the upper tail probability, p , associated with the one-sided Kolmogorov–Smirnov test statistic D_n^+ or D_n^- , where these one-sided statistics are defined as follows;

$$D_n^+ = \sup_x [S_n(x) - F_0(x)],$$

$$D_n^- = \sup_x [F_0(x) - S_n(x)].$$

If $n \leq 100$ an exact method is used; for the details see Conover (1980). Otherwise a large sample approximation derived by Smirnov is used; see Feller (1948), Kendall and Stuart (1973) or Smirnov (1948).

4 References

Conover W J (1980) *Practical Nonparametric Statistics* Wiley

Feller W (1948) On the Kolmogorov–Smirnov limit theorems for empirical distributions *Ann. Math. Statist.* **19** 179–181

Kendall M G and Stuart A (1973) *The Advanced Theory of Statistics (Volume 2)* (3rd Edition) Griffin

Siegel S (1956) *Non-parametric Statistics for the Behavioral Sciences* McGraw–Hill

Smirnov N (1948) Table for estimating the goodness of fit of empirical distributions *Ann. Math. Statist.* **19** 279–281

5 Arguments

1: **n** – Integer *Input*

On entry: n , the number of observations in the sample.

Constraint: $n \geq 1$.

2: **d** – double *Input*

On entry: contains the test statistic, D_n^+ or D_n^- .

Constraint: $0.0 \leq d \leq 1.0$.

3: **fail** – NagError *

Input/Output

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

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.

See Section 3.2.1.2 in the Essential Introduction for further information.

NE_INT

On entry, **n** = $\langle value \rangle$.

Constraint: **n** ≥ 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 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_REAL

On entry, **d** < 0.0 or **d** > 1.0 : **d** = $\langle value \rangle$.

7 Accuracy

The large sample distribution used as an approximation to the exact distribution should have a relative error of less than 2.5% for most cases.

8 Parallelism and Performance

Not applicable.

9 Further Comments

The upper tail probability for the two-sided statistic, $D_n = \max(D_n^+, D_n^-)$, can be approximated by twice the probability returned via nag_prob_1_sample_ks (g01eyc), that is $2p$. (Note that if the probability from nag_prob_1_sample_ks (g01eyc) is greater than 0.5 then the two-sided probability should be truncated to 1.0). This approximation to the tail probability for D_n is good for small probabilities, (e.g., $p \leq 0.10$) but becomes very poor for larger probabilities.

The time taken by the function increases with n , until $n > 100$. At this point the approximation is used and the time decreases significantly. The time then increases again modestly with n .

10 Example

The following example reads in 10 different sample sizes and values for the test statistic D_n . The upper tail probability is computed and printed for each case.

10.1 Program Text

```

/* nag_prob_1_sample_ks (g01eyc) Example Program.
 *
 * Copyright 2014 Numerical Algorithms Group.
 *
 * Mark 7, 2001.
 */

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

int main(void)
{
    /* Scalars */
    double d__, prob;
    Integer exit_status, n;
    NagError fail;

    INIT_FAIL(fail);

    exit_status = 0;
    printf("nag_prob_1_sample_ks (g01eyc) Example Program Results\n\n");
    printf("%s\n\n", "      d      n      One-sided probability");

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

#ifdef _WIN32
    while (scanf_s("%"NAG_IFMT" %lf%*[\n] ", &n, &d__) != EOF)
#else
    while (scanf("%"NAG_IFMT" %lf%*[\n] ", &n, &d__) != EOF)
#endif
    {
        /* nag_prob_1_sample_ks (g01eyc).
         * Computes probabilities for the one-sample
         * Kolmogorov-Smirnov distribution
         */
        prob = nag_prob_1_sample_ks(n, d__, &fail);
        if (fail.code != NE_NOERROR)
        {
            printf("Error from nag_prob_1_sample_ks (g01eyc).\n%s\n",
                fail.message);
            exit_status = 1;
            goto END;
        }
        printf("%7.4f%2s%4"NAG_IFMT"%10s%7.4f\n", d__, "", n, "", prob);
    }
    END:
    return exit_status;
}

```

10.2 Program Data

```

nag_prob_1_sample_ks (g01eyc) Example Program Data
10  0.323
10  0.369
10  0.409
10  0.457
10  0.489

```

400 0.0535
400 0.061
400 0.068
400 0.076
400 0.0815

10.3 Program Results

nag_prob_1_sample_ks (g01eyc) Example Program Results

d	n	One-sided probability
0.3230	10	0.0994
0.3690	10	0.0497
0.4090	10	0.0251
0.4570	10	0.0099
0.4890	10	0.0050
0.0535	400	0.1001
0.0610	400	0.0502
0.0680	400	0.0243
0.0760	400	0.0096
0.0815	400	0.0048
