1 Purpose

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

2 Specification

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

Statist. 19 179–181
Smirnov N (1948) Table for estimating the goodness of fit of empirical distributions Ann. Math. Statist. 19
279–281

5 Parameters

1:  n – Integer

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

   Constraint: $n \geq 1$.

2:  d – double

   On entry: contains the test statistic, $D^+_n$ or $D^-_n$.

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

3:  fail – NagError *

   The NAG error parameter (see the Essential Introduction).
6 Error Indicators and Warnings

NE_INT

On entry, \( n = (\text{value}) \).

Constraint: \( n \geq 1 \).

NE_REAL

On entry, \( d < 0.0 \) or \( d > 1.0 \): \( d = (\text{value}) \).

NE_BAD_PARAM

On entry, parameter \( (\text{value}) \) 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 consult NAG for assistance.

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 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 \textit{nag_prob_1_sample_ks} (g01eyc), that is \( 2p \). (Note that if the probability from \textit{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 \textit{nag_prob_1_sample_ks} (g01eyc) 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 \( n \).

9 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.

9.1 Program Text

```c
#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>

int main(void)
{

  /* Scalars */
  double d__, prob;
  Integer exit_status, n;
  NagError fail;

  INIT_FAIL(fail);
```

/* nag_prob_1_sample_ks (g01eyc) Example Program.
 * Copyright 2001 Numerical Algorithms Group.
 */

```c
#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>

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

  INIT_FAIL(fail);
```
exit_status = 0;
Vprintf("\%s\n\n", "g0leyc Example Program Results");
Vprintf("\%s\n\n", " d n One-sided probability");

/* Skip heading in data file */
Vscanf("%*[\n"]");

while (scanf("%ld %lf%*[\n"] , &n, &d__) != EOF)
{
    prob = g0leyc(n, d__, &fail);
    if (fail.code != NE_NOERROR)
    {
        Vprintf("Error from g0leyc.\n\n", fail.message);
        exit_status = 1;
        goto END;
    }
    Vprintf("%7.4f%2s%4ld%10s%7.4f\n", d__, "", n, "", prob);
}

END:
return exit_status;
}

9.2 Program Data

g0leyc 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

9.3 Program Results

g0leyc 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