NAG Library Function Document

nag_prob_2_sample_ks (g01ezc)

1 Purpose

nag_prob_2_sample_ks (g01ezc) returns the probability associated with the upper tail of the Kolmogorov–Smirnov two sample distribution.

2 Specification

```c
#include <nag.h>
#include <nagg01.h>
double nag_prob_2_sample_ks (Integer n1, Integer n2, double d,
   NagError *fail)
```

3 Description

Let $F_{n_1}(x)$ and $G_{n_2}(x)$ denote the empirical cumulative distribution functions for the two samples, where $n_1$ and $n_2$ are the sizes of the first and second samples respectively.

The function nag_prob_2_sample_ks (g01ezc) computes the upper tail probability for the Kolmogorov–Smirnov two sample two-sided test statistic $D_{n_1,n_2}$, where

$$D_{n_1,n_2} = \sup_x |F_{n_1}(x) - G_{n_2}(x)|.$$  

The probability is computed exactly if $n_1, n_2 \leq 10000$ and $\max(n_1, n_2) \leq 2500$ using a method given by Kim and Jenrich (1973). For the case where $\min(n_1, n_2) \leq 10\%$ of the $\max(n_1, n_2)$ and $\min(n_1, n_2) \leq 80$ the Smirnov approximation is used. For all other cases the Kolmogorov approximation is used. These two approximations are discussed in Kim and Jenrich (1973).

4 References

- Kim P J and Jenrich R I (1973) Tables of exact sampling distribution of the two sample Kolmogorov–Smirnov criterion $D_{min}(m < n)$ *Selected Tables in Mathematical Statistics* 1 80–129 American Mathematical Society

5 Arguments

1:  
   **n1** – Integer  
   ```
   Input
   
   On entry: the number of observations in the first sample, $n_1$.
   ```
   ```
   Constraint: n1 ≥ 1.
   ```
2: \texttt{n2} – Integer  
\textit{Input}

\textit{On entry:} the number of observations in the second sample, \(n_2\).

\textit{Constraint:} \(n_2 \geq 1\).

3: \texttt{d} – double  
\textit{Input}

\textit{On entry:} the test statistic \(D_{n_1,n_2}\), for the two sample Kolmogorov–Smirnov goodness-of-fit test, that is the maximum difference between the empirical cumulative distribution functions (CDFs) of the two samples.

\textit{Constraint:} \(0.0 \leq d \leq 1.0\).

4: \texttt{fail} – NagError*  
\textit{Input/Output}

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

6 \hspace{1em} \textbf{Error Indicators and Warnings}

\textbf{NE_ALLOC_FAIL}

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

\textbf{NE_CONVERGENCE}

The Smirnov approximation used for large samples did not converge in 200 iterations. The probability is set to 1.0.

\textbf{NE_INT}

On entry, \(n_1 = \langle\text{value}\rangle\) and \(n_2 = \langle\text{value}\rangle\).

\textit{Constraint:} \(n_1 \geq 1\) and \(n_2 \geq 1\).

\textbf{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.

\textbf{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.

\textbf{NE_REAL}

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

7 \hspace{1em} \textbf{Accuracy}

The large sample distributions used as approximations to the exact distribution should have a relative error of less than 5\% for most cases.

8 \hspace{1em} \textbf{Parallelism and Performance}

Not applicable.
9 Further Comments

The upper tail probability for the one-sided statistics, \( D_{n_1,n_2}^+ \) or \( D_{n_1,n_2}^- \), can be approximated by halving the two-sided upper tail probability returned by \texttt{nag_prob_2_sample_ks (g01ezc)} , that is \( p/2 \). This approximation to the upper tail probability for either \( D_{n_1,n_2}^+ \) or \( D_{n_1,n_2}^- \) is good for small probabilities, (e.g., \( p \leq 0.10 \)) but becomes poor for larger probabilities.

The time taken by the function increases with \( n_1 \) and \( n_2 \), until \( n_1n_2 > 10000 \) or \( \max(n_1,n_2) \geq 2500 \). At this point one of the approximations is used and the time decreases significantly. The time then increases again modestly with \( n_1 \) and \( n_2 \).

10 Example

The following example reads in 10 different sample sizes and values for the test statistic \( D_{n_1,n_2} \). The upper tail probability is computed and printed for each case.

10.1 Program Text

```c
/* nag_prob_2_sample_ks (g01ezc) Example Program. */
* Copyright 2014 Numerical Algorithms Group.
*/
#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagg01.h>

int main(void)
{
    /* Scalars */
    double d__, prob;
    Integer exit_status, n1, n2;
    NagError fail;

    INIT_FAIL(fail);

    exit_status = 0;
    printf("nag_prob_2_sample_ks (g01ezc) Example Program Results\n\n");
    printf(" d n1 n2 Two-sided probability\n\n");

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

    #ifdef _WIN32
        while (scanf_s("%"NAG_IFMT"%"NAG_IFMT"%f*[\n ] ", &n1, &n2, &d__) != EOF)
            #endif
    #else
        while (scanf("%"NAG_IFMT"%"NAG_IFMT"%f*[\n ] ", &n1, &n2, &d__) != EOF)
            #endif
    {
        /* nag_prob_2_sample_ks (g01ezc). */
        * Computes probabilities for the two-sample
        * Kolmogorov Smirnov distribution */
        prob = nag_prob_2_sample_ks(n1, n2, d__, &fail);
        if (fail.code != NE_NOERROR)
            { printf("Error from nag_prob_2_sample_ks (g01ezc).\n\n", fail.message);
```
exit_status = 1;
goto END;
}
printf("%7.4f%2s%4"NAG_IFMT"%2s%4"NAG_IFMT"%10s%7.4f
", d__,
  
"
", n1, 
"
", n2, 
"
", prob);
END:
  return exit_status;
}

10.2 Program Data

nag_prob_2_sample_ks (g01ezc) Example Program Data
  5 10 0.5
  10 10 0.5
  20 10 0.5
  20 15 0.4833
  400 200 0.1412
  200 20 0.2861
  1000 20 0.2113
  200 50 0.1796
  15 200 0.18
  100 100 0.18

10.3 Program Results

nag_prob_2_sample_ks (g01ezc) Example Program Results

d  n1  n2  Two-sided probability
  0.5000  5  10  0.3506
  0.5000  10 10  0.1678
  0.5000  20 10  0.0623
  0.4833  20 15  0.0261
  0.1412 400 200  0.0083
  0.2861 200 20  0.0789
  0.2113 1000 20  0.2941
  0.1796 200 50  0.1392
  0.1800  15 200  0.6926
  0.1800 100 100  0.0782