NAG C Library Function Document

nag_2_sample_ks_test (g08cdc)

1 Purpose

nag_2_sample_ks_test (g08cdc) performs the two sample Kolmogorov–Smirnov distribution test.

2 Specification

```
#include <nag.h>
#include <nagg08.h>

void nag_2_sample_ks_test (Integer n1, const double x[], Integer n2,
          const double y[], Nag_TestStatistics dtype, double *d, double *z,
          double *p, NagError *fail)
```

3 Description

The data consist of two independent samples, one of size n_1, denoted by $x_1, x_2, \ldots, x_{n_1}$, and the other of size n_2 denoted by $y_1, y_2, \ldots, y_{n_2}$. Let $F(x)$ and $G(x)$ represent their respective, unknown, distribution functions. Also let $S_1(x)$ and $S_2(x)$ denote the values of the sample cumulative distribution functions at the point x for the two samples respectively.

The Kolmogorov–Smirnov test provides a test of the null hypothesis $H_0 : F(x) = G(x)$ against one of the following alternative hypotheses:

(i) $H_1 : F(x) \neq G(x)$.

(ii) $H_2 : F(x) > G(x)$. This alternative hypothesis is sometimes stated as, 'The x's tend to be smaller than the y's', i.e., it would be demonstrated in practical terms if the values of $S_1(x)$ tended to exceed the corresponding values of $S_2(x)$.

(iii) $H_3 : F(x) < G(x)$. This alternative hypothesis is sometimes stated as, 'The x's tend to be larger than the y's', i.e., it would be demonstrated in practical terms if the values of $S_2(x)$ tended to exceed the corresponding values of $S_1(x)$.

One of the following test statistics is computed depending on the particular alternative null hypothesis specified (see the description of the parameter **dtype** in Section 4).

For the alternative hypothesis H_1.

D_{n_1,n_2} – the largest absolute deviation between the two sample cumulative distribution functions.

For the alternative hypothesis H_2.

$D^+_{n_1,n_2}$ – the largest positive deviation between the sample cumulative distribution function of the first sample, $S_1(x)$, and the sample cumulative distribution function of the second sample, $S_2(x)$. Formally $D^+_{n_1,n_2} = \max\{S_1(x) - S_2(x), 0\}$.

For the alternative hypothesis H_3.

$D^-_{n_1,n_2}$ – the largest positive deviation between the sample cumulative distribution function of the second sample, $S_2(x)$, and the sample cumulative distribution function of the first sample, $S_1(x)$. Formally $D^-_{n_1,n_2} = \max\{S_2(x) - S_1(x), 0\}$.

nag_2_sample_ks_test also returns the standardized statistic $Z = \sqrt{(n_1 + n_2/n_1 n_2)} \times D$ where D may be D_{n_1,n_2}, $D^+_{n_1,n_2}$ or $D^-_{n_1,n_2}$ depending on the choice of the alternative hypothesis. The distribution of this statistic converges asymptotically to a distribution given by Smirnov as n_1 and n_2 increase (see Feller (1948), Kendall and Stuart (1973), Kim and Jenrich (1973), Smirnov (1933) or Smirnov (1948)).

The probability, under the null hypothesis, of obtaining a value of the test statistic as extreme as that observed, is computed. If $\max(n_1, n_2) \leq 2500$ and $n_1 n_2 \leq 10000$ then an exact method given by Kim and Jenrich is used. Otherwise p is computed using the approximations suggested by Kim and Jenrich (see Kim and Jenrich (1973)). Note that the method used is only exact for continuous theoretical distributions. This method computes the two-sided probability. The one-sided probabilities are estimated by halving the two-sided probability. This is a good estimate for small p, that is $p \leq 0.10$, but it becomes very poor for larger p.

4 Parameters

1: **n1** – Integer *Input*

On entry: the number of observations in the first sample, n_1.

Constraint: **n1** ≥ 1.

2: **x[n1]** – const double *Input*

On entry: the observations from the first sample, $x_1, x_2, \ldots, x_{n_1}$.

3: **n2** – Integer *Input*

On entry: the number of observations in the second sample, n_2.

Constraint: **n2** ≥ 1.

4: **y[n2]** – const double *Input*

On entry: the observations from the second sample, $y_1, y_2, \ldots, y_{n_2}$.

5: **dtype** – Nag_TestStatistics *Input*

On entry: the statistic to be computed, i.e., the choice of alternative hypothesis.

dtype $=$ **Nag_TestStatisticsDAbs** : computes $D_{n_1 n_2}$, to test against H_1.

dtype $=$ **Nag_TestStatisticsDPos** : computes $D^+_{n_1 n_2}$, to test against H_2.

dtype $=$ **Nag_TestStatisticsDNeg** : computes $D^-_{n_1 n_2}$, to test against H_3.

Constraint: **dtype** $=$ **Nag_TestStatisticsDAbs**, **Nag_TestStatisticsDPos** or **Nag_TestStatisticsDNeg**.

6: **d** – double * *Output*

On exit: the Kolmogorov–Smirnov test statistic ($D_{n_1 n_2}$, $D^+_{n_1 n_2}$ or $D^-_{n_1 n_2}$ according to the value of **dtype**).

7: **z** – double * *Output*

On exit: a standardized value, Z, of the test statistic, D, without any correction for continuity.

8: **p** – double * *Output*

On exit: the tail probability associated with the observed value of D, where D may be D_{n_1,n_2}, $D^+_{n_1,n_2}$ or $D^-_{n_1,n_2}$ depending on the value of **dtype** (see Section 3).

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

The NAG error parameter (see the Essential Introduction).

5 Error Indicators and Warnings

NE_INT_ARG_LT

On entry, **n1** must not be less than 1: **n1** = <*value*>.
On entry, **n2** must not be less than 1: **n2** = <*value*>.

NE_BAD_PARAM

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

NE_G08CD_CONV

The iterative procedure used in the approximation of the probability for large **n1** and **n2** did not converge. For the two-sided test, **p** = 1 is returned. For the one-sided test, **p** = 0.5 is returned.

NE_ALLOC_FAIL

Memory allocation failed.

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.

6 Further Comments

The time taken by the routine increases with n_1 and n_2, until $n_1 n_2 > 10000$ 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.

6.1 Accuracy

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

6.2 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)* Griffin (3rd Edition)

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

Smirnov N (1933) Estimate of deviation between empirical distribution functions in two independent samples *Bull. Moscow Univ.* **2 (2)** 3–16

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

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

7 See Also

None.

8 Example

The following example computes the two-sided Kolmogorov–Smirnov test statistic for two independent samples of size 100 and 50 respectively. The first sample is from a uniform distribution $U(0,2)$. The second sample is from a uniform distribution $U(0.25,2.25)$. The test statistic, D_{n_1,n_2}, the standardized test statistic, Z, and the tail probability, p, are computed and printed.

8.1 Program Text

```
/* nag_2_sample_ks_test (g08cdc) Example Program.
 *
 * Copyright 2000 Numerical Algorithms Group.
 *
 * Mark 6, 2000.
 */


#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagg05.h>
#include <nagg08.h>

int main (void)
{
  double d, enda, endb, p, *x=0, *y=0, z;
  Integer init, i, m, n, ntype;
  Integer exit_status=0;
  NagError fail;
  Nag_TestStatistics ntype_enum;

  INIT_FAIL(fail);
  Vprintf("g08cdc Example Program Results\n");

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

  Vscanf("%ld %ld", &n, &m);
  if (!(x = NAG_ALLOC(n, double))
      || !(y = NAG_ALLOC(m, double)))
    {
      Vprintf("Allocation failure\n");
      exit_status = -1;
      goto END;
    }
  Vprintf("\n");
  init = 0;
  g05cbc(init);
  enda = 0.0;
  endb = 2.0;
  for (i=0;i<n;i++)
    x[i]=enda + (endb-enda) * g05cac();
  enda = 0.25;
  endb = 2.25;
  for (i=0;i<m;i++)
    y[i]=enda + (endb-enda) * g05cac();

  Vscanf("%ld", &ntype);
  if (ntype == 1)
```

```
    ntype_enum =  Nag_TestStatisticsDAbs;
  else if (ntype == 2)
    ntype_enum =  Nag_TestStatisticsDPos;
  else if (ntype == 3)
    ntype_enum =  Nag_TestStatisticsDNeg;
  else
    ntype_enum = (Nag_TestStatistics)-999;

  g08cdc(n, x, m, y, ntype_enum, &d, &z, &p, &fail);
  if (fail.code != NE_NOERROR)
    {
      Vprintf("Error from g08cdc.\n%s\n", fail.message);
      exit_status = 1;
      goto END;
    }
  Vprintf("Test statistic D = %8.4f\n", d);
  Vprintf("Z statistic      = %8.4f\n", z);
  Vprintf("Tail probability = %8.4f\n", p);
 END:
  if (x) NAG_FREE(x);
  if (y) NAG_FREE(y);
  return exit_status;
}
```

8.2 Program Data

```
g08cdc Example Program Data
 100 50
 1
```

8.3 Program Results

```
g08cdc Example Program Results

Test statistic D =   0.3600
Z statistic      =   0.0624
Tail probability =   0.0003
```