Package ‘KSgeneral’

May 7, 2026

Type Package

Version 2.0.2

Title Computing P-Values of the One-Sample K-S Test and the Two-Sample K-S and Kuiper Tests for (Dis)Continuous Null Distribution

Author Dimitrina S. Dimitrova <D.Dimitrova@city.ac.uk>,
Yun Jia <yunjia2019@gmail.com>,
Vladimir K. Kaishev <Vladimir.Kaishev.1@city.ac.uk>,
Senren Tan <raymondtsrtsr@outlook.com>

Maintainer Dimitrina S. Dimitrova <D.Dimitrova@city.ac.uk>

Depends R (>= 3.3.0)

SystemRequirements fftw3 (>=3.3.4)

Copyright Copyright holders of FFTW3: Copyright (c) 2003, 2007-11 Matteo Frigo; Copyright (c) 2003, 2007-11 Massachusetts Institute of Technology

Description Contains functions to compute p-values for the one-sample and two-sample Kolmogorov-Smirnov (KS) tests and the two-sample Kuiper test for any fixed critical level and arbitrary (possibly very large) sample sizes. For the one-sample KS test, this package implements a novel, accurate and efficient method named Exact-KS-FFT, which allows the pre-specified cumulative distribution function under the null hypothesis to be continuous, purely discrete or mixed. In the two-sample case, it is assumed that both samples come from an unspecified (unknown) continuous, purely discrete or mixed distribution, i.e. ties (repeated observations) are allowed, and exact p-values of the KS and the Kuiper tests are computed. Note, the two-sample Kuiper test is often used when data samples are on the line or on the circle (circular data). To cite this package in publication: (for the use of the one-sample KS test) Dimitrina S. Dimitrova, Vladimir K. Kaishev, and Senren Tan. Computing the Kolmogorov-Smirnov Distribution When the Underlying CDF is Purely Discrete, Mixed, or Continuous. Journal of Statistical Software. 2020; 95(10): 1--42. <doi:10.18637/jss.v095.i10>. (for the use of the two-sample KS and Kuiper tests) Dimitrina S. Dimitrova, Yun Jia and Vladimir K. Kaishev (2024). The R functions KS2sample and Kuiper2sample: Efficient Exact Calculation of P-values of the Two-sample Kolmogorov-Smirnov and Kuiper Tests. submitted.

License GPL (>= 2.0)

URL https://github.com/d-dimitrova/KSgeneral

Encoding UTF-8

LazyData true

Imports Rcpp (>= 0.12.12), MASS, dgof

LinkingTo Rcpp

NeedsCompilation yes

Repository CRAN

Date/Publication 2024-07-29 17:20:12 UTC

Contents

KSgeneral-package . . . 2
cont_ks_cdf . . . 5
cont_ks_c_cdf . . . 7
cont_ks_test . . . 9
disc_ks_c_cdf . . . 10
disc_ks_test . . . 13
KS2sample . . . 16
KS2sample_c_Rcpp . . . 19
KS2sample_Rcpp . . . 22
ks_c_cdf_Rcpp . . . 25
Kuiper2sample . . . 27
Kuiper2sample_c_Rcpp . . . 29
Kuiper2sample_Rcpp . . . 31
mixed_ks_c_cdf . . . 33
mixed_ks_test . . . 35
Population_Data . . . 38

Index **40**

`KSgeneral-package` *Computing P-Values of the One-Sample K-S Test and the Two-Sample K-S and Kuiper Tests for (Dis)Continuous Null Distribution*

Description

This package computes p-values of the one-sample and two-sample Kolmogorov-Smirnov (KS) tests and the two-sample Kuiper test.

The one-sample two-sided Kolmogorov-Smirnov (KS) statistic is one of the most popular goodness-of-fit test statistics that is used to measure how well the distribution of a random sample agrees with a prespecified theoretical distribution. Given a random sample $\{X_1, ..., X_n\}$ of size n with an empirical cdf $F_n(x)$, the two-sided KS statistic is defined as $D_n = \sup |F_n(x) - F(x)|$, where $F(x)$ is the cdf of the prespecified theoretical distribution under the null hypothesis H_0, that $\{X_1, ..., X_n\}$ comes from $F(x)$. The package **KSgeneral** implements a novel, accurate and efficient Fast Fourier Transform (FFT)-based method, referred as Exact-KS-FFT method to compute the complementary cdf, $P(D_n \geq q)$, at a fixed $q \in [0, 1]$ for a given (hypothezied) purely discrete, mixed or continuous

underlying cdf $F(x)$, and arbitrary, possibly very large sample size n. A plot of the complementary cdf $P(D_n \geq q)$, $0 \leq q \leq 1$, can also be produced.

In other words, the package computes the p-value, $P(D_n \geq q)$ for any fixed critical level $q \in [0, 1]$. If an observed (data) sample, $\{x_1, ..., x_n\}$ is supplied, **KSgeneral** computes the p-value $P(D_n \geq d_n)$, where d_n is the value of the KS test statistic computed based on $\{x_1, ..., x_n\}$. One can also compute the (complementary) cdf for the one-sided KS statistics D_n^- or D_n^+ (cf., Dimitrova, Kaishev, Tan (2020)) by appropriately specifying correspondingly $A_i = 0$ for all i or $B_i = 1$ for all i, in the function `ks_c_cdf_Rcpp`.

The two-sample Kolmogorov-Smirnov (KS) and the Kuiper statistics are widely used to test the null hypothesis (H_0) that two data samples come from the same underlying distribution. Given a pair of random samples $\boldsymbol{X}_m = (X_1, ..., X_m)$ and $\boldsymbol{Y}_n = (Y_1, ..., Y_n)$ of sizes `m` and `n` with empirical cdfs $F_m(t)$ and $G_n(t)$ respectively, coming from unknown CDFs $F(x)$ and $G(x)$. It is assumed that $F(x)$ and $G(x)$ could be either *continuous*, *discrete* or *mixed*, which means that repeated observations are allowed in the corresponding observed samples. We want to test the null hypothesis $H_0 : F(x) = G(x)$ for all x, either against the alternative hypothesis $H_1 : F(x) \neq G(x)$ for at least one x, which corresponds to the two-sided test, or against $H_1 : F(x) > G(x)$ and $H_1 : F(x) < G(x)$ for at least one x, which corresponds to the two one-sided tests. The (weighted) two-sample Kolmogorov-Smirnov goodness-of-fit statistics that are used to test these hypotheses are generally defined as:

$$\Delta_{m,n} = \sup |F_m(t) - G_n(t)| W(E_{m+n}(t), \text{ to test against the alternative } H_1 : F(x) \neq G(x)$$

$$\Delta^+_{m,n} = \sup [F_m(t) - G_n(x)] W(E_{m+n}(t)), \text{ to test against the alternative } H_1 : F(x) > G(x)$$

$$\Delta^-_{m,n} = \sup [G_n(t) - F_m(x)] W(E_{m+n}(t)), \text{ to test against the alternative } H_1 : F(x) < G(x)$$

where $E_{m+n}(t)$ is the empirical cdf of the pooled sample $\boldsymbol{Z}_{m,n} = (X_1, ..., X_m, Y_1, ..., Y_n)$, $W()$ is a strictly positive weight function defined on $(0, 1)$. **KSgeneral** implements an exact algorithm which is an extension of the Fortran 77 subroutine due to Nikiforov (1994), to calculate the exact p-value $P(D_{m,n} \geq q)$, where $q \in [0, 1]$ and $D_{m,n}$ is the two-sample Kolmogorov-Smirnov goodness-of-fit test defined on the space Ω of all possible $\frac{(m+n)!}{m!n!}$ pairs of samples, $\boldsymbol{X}'_m$ and $\boldsymbol{Y}'_n$ of sizes m and n, that are *randomly drawn from the pooled sample* $\boldsymbol{Z}_{m+n}$ *without replacement*. If two data samples $\{x_1, \ldots, x_m\}$ and $\{y_1, \ldots, y_n\}$ are supplied, the package computes $P(D_{m,n} \geq \mathsf{d})$, where d is the observed value of $\Delta_{m,n}$ computed based on these two observed samples. Samples may come from any continuous, discrete or mixed distribution, i.e. the test allows repeated observations to appear in the user provided data samples $\{x_1, \ldots, x_m\}$, $\{y_1, \ldots, y_n\}$ and their pooled sample $\boldsymbol{Z}_{m+n} = \{x_1, \ldots, x_m, y_1, \ldots, y_n\}$.

The two-sample (unweighted) Kuiper goodness-of-fit statistic is defined as:

$$\varsigma_{m,n} = \sup[F_m(t) - G_n(t)] - \inf[F_m(t) - G_n(t)].$$

It is widely used when the data samples are periodic or circular (data that are measured in radians). **KSgeneral** calculates the exact p-value $P(V_{m,n} \geq q)$, where $q \in [0, 2]$ and $V_{m,n}$ is the two-sample Kuiper goodness-of-fit test defined on the on the space, Ω, as described above. If two data samples $\{x_1, \ldots, x_m\}$ and $\{y_1, \ldots, y_n\}$ are supplied, the package computes $P(V_{m,n} \geq \mathsf{v})$, where v is the observed value of $\varsigma_{m,n}$ computed based on these two observed samples. Similarly, as for the KS test, the two-sample Kuiper test also allows repeated observations in the user provided data samples $\{x_1, \ldots, x_m\}$, $\{y_1, \ldots, y_n\}$ and their pooled sample $\boldsymbol{Z}_{m+n} = \{x_1, \ldots, x_m, y_1, \ldots, y_n\}$.

Details

One-sample KS test:

The Exact-KS-FFT method to compute p-values of the one-sample KS test in **KSgeneral** is based on expressing the p-value $P(D_n \geq q)$ in terms of an appropriate rectangle probability with respect to the uniform order statistics, as noted by Gleser (1985) for $P(D_n > q)$. The latter representation is used to express $P(D_n \geq q)$ via a double-boundary non-crossing probability for a homogeneous Poisson process, with intensity n, which is then efficiently computed using FFT, ensuring total run-time of order $O(n^2 log(n))$ (see Dimitrova, Kaishev, Tan (2020) and also Moscovich and Nadler (2017) for the special case when $F(x)$ is continuous).

The code for the one-sample KS test in **KSgeneral** represents an R wrapper of the original C++ code due to Dimitrova, Kaishev, Tan (2020) and based on the C++ code developed by Moscovich and Nadler (2017). The package includes the functions `disc_ks_c_cdf`, `mixed_ks_c_cdf` and `cont_ks_c_cdf` that compute the complementary cdf $P(D_n \geq q)$, for a fixed q, $0 \leq q \leq 1$, when $F(x)$ is purely discrete, mixed or continuous, respectively. **KSgeneral** includes also the functions `disc_ks_test`, `mixed_ks_test` and `cont_ks_test` that compute the p-value $P(D_n \geq d_n)$, where d_n is the value of the KS test statistic computed based on a user provided data sample $\{x_1, ..., x_n\}$, when $F(x)$ is purely discrete, mixed or continuous, respectively.

The functions `disc_ks_test` and `cont_ks_test` represent accurate and fast (run time $O(n^2 log(n))$) alternatives to the functions `ks.test` from the package **dgof** and the function `ks.test` from the package **stat**, which compute p-values of $P(D_n \geq d_n)$, assuming $F(x)$ is purely discrete or continuous, respectively.

The package also includes the function `ks_c_cdf_Rcpp` which gives the flexibility to compute the complementary cdf (p-value) for the one-sided KS test statistics D_n^- or D_n^+. It also allows for faster computation time and possibly higher accuracy in computing $P(D_n \geq q)$.

Two-sample KS test and Kuiper test:

The method underlying for computing p-values of the two-sample KS and Kuiper tests in **KSgeneral** is the extension of the algorithm due to Nikiforov (1994) and is based on expressing the p-value as the probability that a point sequence stays within a certain region in the two-dimensional integer-valued lattice. The algorithm for both tests uses a recursive formula to calculate the total number of point sequences within the region which is divided by the total number of elements in Ω, i.e. $\frac{(m+n)!}{m!n!}$ to obtain the probability.

For a particular realization of the pooled sample $\boldsymbol{Z}_{m,n} = (X_1, ..., X_m, Y_1, ..., Y_n)$, the p-values calculated by the functions `KS2sample` and `Kuiper2sample` are the probabilities:

$$P(D_{m,n} \geq q), P(V_{m,n} \geq q),$$

where $D_{m,n}$ and $V_{m,n}$ are the two-sample Kolmogorov-Smirnov and Kuiper test statistics respectively, for two samples $\boldsymbol{X}'_m$ and $\boldsymbol{Y}'_n$ of sizes m and n, *randomly drawn from the pooled sample without replacement*, i.e. they are defined on the space Ω and $q \in [0, 1]$ for the KS test, $q \in [0, 2]$ for the Kuiper test.

Both `KS2sample` and `Kuiper2sample` implement algorithms which generalize the method due to Nikiforov (1994), and calculate the exact p-values of the KS test and the Kuiper test respectively. Both of them allow tested data samples to come from continuous, discrete or mixed distributions (ties are also allowed).

`KS2sample` ensures a total worst-case run-time of order $O(nm)$. Compared with other known algorithms, it not only allows more flexible choices on weights leading to better power (see Dimitrova,

Jia, Kaishev 2024), but also is more efficient and more generally applicable for *large sample sizes*. `Kuiper2sample` is accurate and valid for large sample sizes. It ensures a total worst-case run-time of order $O((mn)^2)$. When `m` and `n` have large greatest common divisor (an extreme case is `m = n`), it ensures a total worst-case run-time of order $O((m)^2 n)$.

Author(s)

Dimitrina S. Dimitrova <D.Dimitrova@city.ac.uk>, Yun Jia <yunjia2019@gmail.com>, Vladimir K. Kaishev <Vladimir.Kaishev.1@city.ac.uk>, Senren Tan <raymondtsrtsr@outlook.com>

Maintainer: Dimitrina S. Dimitrova <D.Dimitrova@city.ac.uk>

References

Dimitrina S. Dimitrova, Vladimir K. Kaishev, Senren Tan. (2020) "Computing the Kolmogorov-Smirnov Distribution When the Underlying CDF is Purely Discrete, Mixed or Continuous". Journal of Statistical Software, **95**(10): 1-42. doi:10.18637/jss.v095.i10.

Gleser L.J. (1985). "Exact Power of Goodness-of-Fit Tests of Kolmogorov Type for Discontinuous Distributions". Journal of the American Statistical Association, **80**(392), 954-958.

Moscovich A., Nadler B. (2017). "Fast Calculation of Boundary Crossing Probabilities for Poisson Processes". Statistics and Probability Letters, **123**, 177-182.

Dimitrina S. Dimitrova, Yun Jia, Vladimir K. Kaishev (2024). "The R functions KS2sample and Kuiper2sample: Efficient Exact Calculation of P-values of the Two-sample Kolmogorov-Smirnov and Kuiper Tests". *submitted*

`cont_ks_cdf` *Computes the cumulative distribution function of the two-sided Kolmogorov-Smirnov statistic when the cdf under the null hypothesis is continuous*

Description

Computes the cdf $P(D_n \leq q) \equiv P(D_n < q)$ at a fixed q, $q \in [0, 1]$, for the one-sample two-sided Kolmogorov-Smirnov statistic, D_n, for a given sample size n, when the cdf $F(x)$ under the null hypothesis is continuous.

Usage

```
cont_ks_cdf(q, n)
```

Arguments

`q`	numeric value between 0 and 1, at which the cdf $P(D_n \leq q)$ is computed
`n`	the sample size

Details

Given a random sample $\{X_1, ..., X_n\}$ of size n with an empirical cdf $F_n(x)$, the Kolmogorov-Smirnov goodness-of-fit statistic is defined as $D_n = \sup |F_n(x) - F(x)|$, where $F(x)$ is the cdf of a prespecified theoretical distribution under the null hypothesis H_0, that $\{X_1, ..., X_n\}$ comes from $F(x)$.

The function cont_ks_cdf implements the FFT-based algorithm proposed by Moscovich and Nadler (2017) to compute the cdf $P(D_n \leq q)$ at a value q, when $F(x)$ is continuous. This algorithm ensures a total worst-case run-time of order $O(n^2 log(n))$ which makes it more efficient and numerically stable than the algorithm proposed by Marsaglia et al. (2003). The latter is used by many existing packages computing the cdf of D_n, e.g., the function ks.test in the package **stats** and the function ks.test in the package **dgof**. More precisely, in these packages, the exact p-value, $P(D_n \geq q)$ is computed only in the case when $q = d_n$, where d_n is the value of the KS statistic computed based on a user provided sample $\{x_1, ..., x_n\}$. Another limitation of the functions ks.test is that the sample size should be less than 100, and the computation time is $O(n^3)$. In contrast, the function cont_ks_cdf provides results with at least 10 correct digits after the decimal point for sample sizes n up to 100000 and computation time of 16 seconds on a machine with an 2.5GHz Intel Core i5 processor with 4GB RAM, running MacOS X Yosemite. For n > 100000, accurate results can still be computed with similar accuracy, but at a higher computation time. See Dimitrova, Kaishev, Tan (2020), Appendix B for further details and examples.

Value

Numeric value corresponding to $P(D_n \leq q)$.

Source

Based on the C++ code available at https://github.com/mosco/crossing-probability developed by Moscovich and Nadler (2017). See also Dimitrova, Kaishev, Tan (2020) for more details.

References

Dimitrina S. Dimitrova, Vladimir K. Kaishev, Senren Tan. (2020) "Computing the Kolmogorov-Smirnov Distribution When the Underlying CDF is Purely Discrete, Mixed or Continuous". Journal of Statistical Software, **95**(10): 1-42. doi:10.18637/jss.v095.i10.

Marsaglia G., Tsang WW., Wang J. (2003). "Evaluating Kolmogorov's Distribution". Journal of Statistical Software, **8**(18), 1-4.

Moscovich A., Nadler B. (2017). "Fast Calculation of Boundary Crossing Probabilities for Poisson Processes". Statistics and Probability Letters, **123**, 177-182.

Examples

```
## Compute the value for P(D_{100} <= 0.05)

KSgeneral::cont_ks_cdf(0.05, 100)


## Compute P(D_{n} <= q)
## for n = 100, q = 1/500, 2/500, ..., 500/500
```

```
## and then plot the corresponding values against q

n<-100
q<-1:500/500
plot(q, sapply(q, function(x) KSgeneral::cont_ks_cdf(x, n)), type='l')

## Compute P(D_{n} <= q) for n = 40, nq^{2} = 0.76 as shown
## in Table 9 of Dimitrova, Kaishev, Tan (2020)

KSgeneral::cont_ks_cdf(sqrt(0.76/40), 40)
```

`cont_ks_c_cdf` *Computes the complementary cumulative distribution function of the two-sided Kolmogorov-Smirnov statistic when the cdf under the null hypothesis is continuous*

Description

Computes the complementary cdf $P(D_n \geq q) \equiv P(D_n > q)$ at a fixed q, $q \in [0, 1]$, for the one-sample two-sided Kolmogorov-Smirnov statistic, D_n, for a given sample size n, when the cdf $F(x)$ under the null hypothesis is continuous.

Usage

```
cont_ks_c_cdf(q, n)
```

Arguments

q	numeric value between 0 and 1, at which the complementary cdf $P(D_n \geq q)$ is computed
n	the sample size

Details

Given a random sample $\{X_1, ..., X_n\}$ of size n with an empirical cdf $F_n(x)$, the two-sided Kolmogorov-Smirnov goodness-of-fit statistic is defined as $D_n = \sup |F_n(x) - F(x)|$, where $F(x)$ is the cdf of a prespecified theoretical distribution under the null hypothesis H_0, that $\{X_1, ..., X_n\}$ comes from $F(x)$.

The function `cont_ks_c_cdf` implements the FFT-based algorithm proposed by Moscovich and Nadler (2017) to compute the complementary cdf, $P(D_n \geq q)$ at a value q, when $F(x)$ is continuous. This algorithm ensures a total worst-case run-time of order $O(n^2 log(n))$ which makes it more efficient and numerically stable than the algorithm proposed by Marsaglia et al. (2003). The latter is used by many existing packages computing the cdf of D_n, e.g., the function `ks.test` in the package **stats** and the function `ks.test` in the package **dgof**. More precisely, in these packages, the exact p-value, $P(D_n \geq q)$ is computed only in the case when $q = d_n$, where d_n is the value of the KS test statistic computed based on a user provided sample $\{x_1, ..., x_n\}$. Another limitation of the functions `ks.test` is that the sample size should be less than 100, and the computation time is

$O(n^3)$. In contrast, the function `cont_ks_c_cdf` provides results with at least 10 correct digits after the decimal point for sample sizes n up to 100000 and computation time of 16 seconds on a machine with an 2.5GHz Intel Core i5 processor with 4GB RAM, running MacOS X Yosemite. For `n` > 100000, accurate results can still be computed with similar accuracy, but at a higher computation time. See Dimitrova, Kaishev, Tan (2020), Appendix C for further details and examples.

Value

Numeric value corresponding to $P(D_n \geq q)$.

Source

Based on the C++ code available at `https://github.com/mosco/crossing-probability` developed by Moscovich and Nadler (2017). See also Dimitrova, Kaishev, Tan (2020) for more details.

References

Dimitrina S. Dimitrova, Vladimir K. Kaishev, Senren Tan. (2020) "Computing the Kolmogorov-Smirnov Distribution When the Underlying CDF is Purely Discrete, Mixed or Continuous". Journal of Statistical Software, **95**(10): 1-42. doi:10.18637/jss.v095.i10.

Marsaglia G., Tsang WW., Wang J. (2003). "Evaluating Kolmogorov's Distribution". Journal of Statistical Software, **8**(18), 1-4.

Moscovich A., Nadler B. (2017). "Fast Calculation of Boundary Crossing Probabilities for Poisson Processes". Statistics and Probability Letters, **123**, 177-182.

Examples

```
## Compute the value for P(D_{100} >= 0.05)

KSgeneral::cont_ks_c_cdf(0.05, 100)


## Compute P(D_{n} >= q)
## for n = 100, q = 1/500, 2/500, ..., 500/500
## and then plot the corresponding values against q

n <- 100
q <- 1:500/500
plot(q, sapply(q, function(x) KSgeneral::cont_ks_c_cdf(x, n)), type='l')

## Compute P(D_{n} >= q) for n = 141, nq^{2} = 2.1 as shown
## in Table 18 of Dimitrova, Kaishev, Tan (2020)

KSgeneral::cont_ks_c_cdf(sqrt(2.1/141), 141)
```

`cont_ks_test` *Computes the p-value for a one-sample two-sided Kolmogorov-Smirnov test when the cdf under the null hypothesis is continuous*

Description

Computes the p-value $P(D_n \geq d_n) \equiv P(D_n > d_n)$, where d_n is the value of the KS test statistic computed based on a data sample $\{x_1, ..., x_n\}$, when $F(x)$ is continuous.

Usage

```
cont_ks_test(x, y, ...)
```

Arguments

`x`	a numeric vector of data sample values $\{x_1, ..., x_n\}$.
`y`	a pre-specified continuous cdf, $F(x)$ under the null hypothesis. Note that `y` should be a character string naming a continuous cumulative distribution function such as `pexp`, `pnorm`, etc. Only continuous cdfs are valid!
`...`	values of the parameters of the cdf, $F(x)$ specified (as a character string) by `y`.

Details

Given a random sample $\{X_1, ..., X_n\}$ of size `n` with an empirical cdf $F_n(x)$, the two-sided Kolmogorov-Smirnov goodness-of-fit statistic is defined as $D_n = \sup |F_n(x) - F(x)|$, where $F(x)$ is the cdf of a prespecified theoretical distribution under the null hypothesis H_0, that $\{X_1, ..., X_n\}$ comes from $F(x)$.

The function `cont_ks_test` implements the FFT-based algorithm proposed by Moscovich and Nadler (2017) to compute the p-value $P(D_n \geq d_n)$, where d_n is the value of the KS test statistic computed based on a user provided data sample $\{x_1, ..., x_n\}$, assuming $F(x)$ is continuous. This algorithm ensures a total worst-case run-time of order $O(n^2 log(n))$ which makes it more efficient and numerically stable than the algorithm proposed by Marsaglia et al. (2003). The latter is used by many existing packages computing the cdf of D_n, e.g., the function `ks.test` in the package **stats** and the function `ks.test` in the package **dgof**. A limitation of the functions `ks.test` is that the sample size should be less than 100, and the computation time is $O(n^3)$. In contrast, the function `cont_ks_test` provides results with at least 10 correct digits after the decimal point for sample sizes n up to 100000 and computation time of 16 seconds on a machine with an 2.5GHz Intel Core i5 processor with 4GB RAM, running MacOS X Yosemite. For `n` > 100000, accurate results can still be computed with similar accuracy, but at a higher computation time. See Dimitrova, Kaishev, Tan (2020), Appendix C for further details and examples.

Value

A list with class "htest" containing the following components:

`statistic`	the value of the statistic.

p.value	the p-value of the test.
alternative	"two-sided".
data.name	a character string giving the name of the data.

Source

Based on the C++ code available at https://github.com/mosco/crossing-probability developed by Moscovich and Nadler (2017). See also Dimitrova, Kaishev, Tan (2020) for more details.

References

Dimitrina S. Dimitrova, Vladimir K. Kaishev, Senren Tan. (2020) "Computing the Kolmogorov-Smirnov Distribution When the Underlying CDF is Purely Discrete, Mixed or Continuous". Journal of Statistical Software, **95**(10): 1-42. doi:10.18637/jss.v095.i10.

Moscovich A., Nadler B. (2017). "Fast Calculation of Boundary Crossing Probabilities for Poisson Processes". Statistics and Probability Letters, **123**, 177-182.

Examples

```
## Comparing the p-values obtained by stat::ks.test
## and KSgeneral::cont_ks_test

x<-abs(rnorm(100))
p.kt <- ks.test(x, "pexp", exact = TRUE)$p
p.kt_fft <- KSgeneral::cont_ks_test(x, "pexp")$p
abs(p.kt-p.kt_fft)
```

---

disc_ks_c_cdf *Computes the complementary cumulative distribution function of the two-sided Komogorov-Smirnov statistic when the cdf under the null hypothesis is purely discrete*

Description

Computes the complementary cdf, $P(D_n \geq q)$ at a fixed q, $q \in [0, 1]$, of the one-sample two-sided Kolmogorov-Smirnov (KS) statistic, when the cdf $F(x)$ under the null hypothesis is purely discrete, using the Exact-KS-FFT method expressing the p-value as a double-boundary non-crossing probability for a homogeneous Poisson process, which is then efficiently computed using FFT (see Dimitrova, Kaishev, Tan (2020)). Moreover, for comparison purposes, disc_ks_c_cdf gives, as an option, the possibility to compute (an approximate value for) the asymptotic $P(D_n \geq q)$ using the simulation-based algorithm of Wood and Altavela (1978).

Usage

```
disc_ks_c_cdf(q, n, y, ..., exact = NULL, tol = 1e-08, sim.size = 1e+06, num.sim = 10)
```

Arguments

`q`	numeric value between 0 and 1, at which the complementary cdf $P(D_n \geq q)$ is computed
`n`	the sample size
`y`	a pre-specified discrete cdf, $F(x)$ under the null hypothesis. Note that `y` should be a step function within the class: `stepfun`, of which `ecdf` is a subclass!
`...`	values of the parameters of the cdf, $F(x)$, specified (as a character string) by `y`.
`exact`	logical variable specifying whether one wants to compute exact p-value $P(D_n \geq q)$ using the Exact-KS-FFT method, in which case `exact = TRUE` or wants to compute an approximate p-value $P(D_n \geq q)$ using the simulation-based algorithm of Wood and Altavela (1978), in which case `exact = FALSE`. When `exact = NULL` and `n <= 100000`, the exact $P(D_n \geq q)$ will be computed using the Exact-KS-FFT method. Otherwise, the asymptotic complementary cdf is computed based on Wood and Altavela (1978). By default, `exact = NULL`.
`tol`	the value of ϵ that is used to compute the values of A_i and B_i, $i = 1, ..., n$, as detailed in Step 1 of Section 2.1 in Dimitrova, Kaishev and Tan (2020) (see also (ii) in the Procedure Exact-KS-FFT therein). By default, `tol = 1e-08`. Note that a value of `NA` or `0` will lead to an error!
`sim.size`	the required number of simulated trajectories in order to produce one Monte Carlo estimate (one MC run) of the asymptotic complementary cdf using the algorithm of Wood and Altavela (1978). By default, `sim.size = 1e+06`.
`num.sim`	the number of MC runs, each producing one estimate (based on `sim.size` number of trajectories), which are then averaged in order to produce the final estimate for the asymptotic complementary cdf. This is done in order to reduce the variance of the final estimate. By default, `num.sim = 10`.

Details

Given a random sample $\{X_1, ..., X_n\}$ of size `n` with an empirical cdf $F_n(x)$, the two-sided Kolmogorov-Smirnov goodness-of-fit statistic is defined as $D_n = \sup |F_n(x) - F(x)|$, where $F(x)$ is the cdf of a prespecified theoretical distribution under the null hypothesis H_0, that $\{X_1, ..., X_n\}$ comes from $F(x)$.

The function `disc_ks_c_cdf` implements the Exact-KS-FFT method, proposed by Dimitrova, Kaishev, Tan (2020) to compute the complementary cdf $P(D_n \geq q)$ at a value q, when $F(x)$ is purely discrete. This algorithm ensures a total worst-case run-time of order $O(n^2 log(n))$ which makes it more efficient and numerically stable than the only alternative algorithm developed by Arnold and Emerson (2011) and implemented as the function `ks.test` in the package **dgof**. The latter only computes a p-value $P(D_n \geq d_n)$, corresponding to the value of the KS test statistic d_n computed based on a user provided sample $\{x_1, ..., x_n\}$. More precisely, in the package **dgof** (function `ks.test`), the p-value for a one-sample two-sided KS test is calculated by combining the approaches of Gleser (1985) and Niederhausen (1981). However, the function `ks.test` only provides exact p-values for `n` ≤ 30, since as noted by the authors (see Arnold and Emerson (2011)), when `n` is large, numerical instabilities may occur. In the latter case, `ks.test` uses simulation to approximate p-values, which may be rather slow and inaccurate (see Table 6 of Dimitrova, Kaishev, Tan (2020)).

Thus, making use of the Exact-KS-FFT method, the function disc_ks_c_cdf provides an exact and highly computationally efficient (alternative) way of computing $P(D_n \geq q)$ at a value q, when $F(x)$ is purely discrete.

Lastly, incorporated into the function disc_ks_c_cdf is the MC simulation-based method of Wood and Altavela (1978) for estimating the asymptotic complementary cdf of D_n. The latter method is the default method behind disc_ks_c_cdf when the sample size n is n $\geq$ 100000.

Value

Numeric value corresponding to $P(D_n \geq q)$.

References

Arnold T.A., Emerson J.W. (2011). "Nonparametric Goodness-of-Fit Tests for Discrete Null Distributions". The R Journal, **3**(2), 34-39.

Dimitrina S. Dimitrova, Vladimir K. Kaishev, Senren Tan. (2020) "Computing the Kolmogorov-Smirnov Distribution When the Underlying CDF is Purely Discrete, Mixed or Continuous". Journal of Statistical Software, **95**(10): 1-42. doi:10.18637/jss.v095.i10.

Gleser L.J. (1985). "Exact Power of Goodness-of-Fit Tests of Kolmogorov Type for Discontinuous Distributions". Journal of the American Statistical Association, **80**(392), 954-958.

Niederhausen H. (1981). "Sheffer Polynomials for Computing Exact Kolmogorov-Smirnov and Renyi Type Distributions". The Annals of Statistics, 58-64.

Wood C.L., Altavela M.M. (1978). "Large-Sample Results for Kolmogorov-Smirnov Statistics for Discrete Distributions". Biometrika, **65**(1), 235-239.

See Also

ks.test

Examples

```
## Example to compute the exact complementary cdf for D_{n}
## when the underlying cdf F(x) is a binomial(3, 0.5) distribution,
## as shown in Example 3.4 of Dimitrova, Kaishev, Tan (2020)

binom_3 <- stepfun(c(0:3), c(0,pbinom(0:3,3,0.5)))
KSgeneral::disc_ks_c_cdf(0.05, 400, binom_3)

## Not run:
## Compute P(D_{n} >= q) for n = 100,
## q = 1/5000, 2/5000, ..., 5000/5000, when
## the underlying cdf F(x) is a binomial(3, 0.5) distribution,
## as shown in Example 3.4 of Dimitrova, Kaishev, Tan (2020),
## and then plot the corresponding values against q,
## i.e. plot the resulting complementary cdf of D_{n}

n <- 100
q <- 1:5000/5000
binom_3 <- stepfun(c(0:3), c(0,pbinom(0:3,3,0.5)))
```

```
plot(q, sapply(q, function(x) KSgeneral::disc_ks_c_cdf(x, n, binom_3)), type='l')

## End(Not run)

## Not run:
## Example to compute the asymptotic complementary cdf for D_{n}
## based on Wood and Altavela (1978),
## when the underlying cdf F(x) is a binomial(3, 0.5) distribution,
## as shown in Example 3.4 of Dimitrova, Kaishev, Tan (2020)

binom_3 <- stepfun(c(0: 3), c(0, pbinom(0 : 3, 3, 0.5)))
KSgeneral::disc_ks_c_cdf(0.05, 400, binom_3, exact = FALSE, tol = 1e-08,
sim.size = 1e+06, num.sim = 10)

## End(Not run)
```

`disc_ks_test` *Computes the p-value for a one-sample two-sided Kolmogorov-Smirnov test when the cdf under the null hypothesis is purely discrete*

Description

Computes the p-value $P(D_n \geq d_n)$, where d_n is the value of the KS test statistic computed based on a data sample $\{x_1, ..., x_n\}$, when $F(x)$ is purely discrete, using the Exact-KS-FFT method expressing the p-value as a double-boundary non-crossing probability for a homogeneous Poisson process, which is then efficiently computed using FFT (see Dimitrova, Kaishev, Tan (2020)).

Usage

```
disc_ks_test(x, y, ..., exact = NULL, tol = 1e-08, sim.size = 1e+06, num.sim = 10)
```

Arguments

`x`	a numeric vector of data sample values $\{x_1, ..., x_n\}$.
`y`	a pre-specified discrete cdf, $F(x)$, under the null hypothesis. Note that `y` should be a step function within the class: `stepfun`, of which `ecdf` is a subclass!
`...`	values of the parameters of the cdf, $F(x)$, specified (as a character string) by `y`.
`exact`	logical variable specifying whether one wants to compute exact p-value $P(D_n \geq d_n)$ using the Exact-KS-FFT method, in which case `exact = TRUE` or wants to compute an approximate p-value $P(D_n \geq d_n)$ using the simulation-based algorithm of Wood and Altavela (1978), in which case `exact = FALSE`. When `exact = NULL` and `n <= 100000`, the exact $P(D_n \geq d_n)$ will be computed using the Exact-KS-FFT method. Otherwise, the asymptotic complementary cdf is computed based on Wood and Altavela (1978). By default, `exact = NULL`.

`tol` the value of ϵ that is used to compute the values of A_i and B_i, $i = 1, ..., n$, as detailed in Step 1 of Section 2.1 in Dimitrova, Kaishev and Tan (2020) (see also (ii) in the Procedure Exact-KS-FFT therein). By default, `tol = 1e-08`. Note that a value of `NA` or `0` will lead to an error!

`sim.size` the required number of simulated trajectories in order to produce one Monte Carlo estimate (one MC run) of the asymptotic p-value using the algorithm of Wood and Altavela (1978). By default, `sim.size = 1e+06`.

`num.sim` the number of MC runs, each producing one estimate (based on `sim.size` number of trajectories), which are then averaged in order to produce the final estimate for the asymptotic p-value. This is done in order to reduce the variance of the final estimate. By default, `num.sim = 10`.

Details

Given a random sample $\{X_1, ..., X_n\}$ of size `n` with an empirical cdf $F_n(x)$, the two-sided Kolmogorov-Smirnov goodness-of-fit statistic is defined as $D_n = \sup |F_n(x) - F(x)|$, where $F(x)$ is the cdf of a prespecified theoretical distribution under the null hypothesis H_0, that $\{X_1, ..., X_n\}$ comes from $F(x)$.

The function `disc_ks_test` implements the Exact-KS-FFT method expressing the p-value as a double-boundary non-crossing probability for a homogeneous Poisson process, which is then efficiently computed using FFT (see Dimitrova, Kaishev, Tan (2020)). It represents an accurate and fast (run time $O(n^2 log(n))$) alternative to the function `ks.test` from the package **dgof**, which computes a p-value $P(D_n \geq d_n)$, where d_n is the value of the KS test statistic computed based on a user provided data sample $\{x_1, ..., x_n\}$, assuming $F(x)$ is purely discrete.

In the function `ks.test`, the p-value for a one-sample two-sided KS test is calculated by combining the approaches of Gleser (1985) and Niederhausen (1981). However, the function `ks.test` due to Arnold and Emerson (2011) only provides exact p-values for `n` ≤ 30, since as noted by the authors, when `n` is large, numerical instabilities may occur. In the latter case, `ks.test` uses simulation to approximate p-values, which may be rather slow and inaccurate (see Table 6 of Dimitrova, Kaishev, Tan (2020)).

Thus, making use of the Exact-KS-FFT method, the function `disc_ks_test` provides an exact and highly computationally efficient (alternative) way of computing the p-value $P(D_n \geq d_n)$, when $F(x)$ is purely discrete.

Lastly, incorporated into the function `disc_ks_test` is the MC simulation-based method of Wood and Altavela (1978) for estimating the asymptotic p-value of D_n. The latter method is the default method behind `disc_ks_test` when the sample size `n` is `n` ≥ 100000.

Value

A list with class "htest" containing the following components:

`statistic` the value of the statistic.

`p.value` the p-value of the test.

`alternative` "two-sided".

`data.name` a character string giving the name of the data.

References

Arnold T.A., Emerson J.W. (2011). "Nonparametric Goodness-of-Fit Tests for Discrete Null Distributions". The R Journal, **3**(2), 34-39.

Dimitrina S. Dimitrova, Vladimir K. Kaishev, Senren Tan. (2020) "Computing the Kolmogorov-Smirnov Distribution When the Underlying CDF is Purely Discrete, Mixed or Continuous". Journal of Statistical Software, **95**(10): 1-42. doi:10.18637/jss.v095.i10.

Gleser L.J. (1985). "Exact Power of Goodness-of-Fit Tests of Kolmogorov Type for Discontinuous Distributions". Journal of the American Statistical Association, **80**(392), 954-958.

Niederhausen H. (1981). "Sheffer Polynomials for Computing Exact Kolmogorov-Smirnov and Renyi Type Distributions". The Annals of Statistics, 58-64.

Wood C.L., Altavela M.M. (1978). "Large-Sample Results for Kolmogorov-Smirnov Statistics for Discrete Distributions". Biometrika, **65**(1), 235-239.

See Also

ks.test

Examples

```
# Comparison of results obtained from dgof::ks.test
# and KSgeneral::disc_ks_test, when F(x) follows the discrete
# Uniform[1, 10] distribution as in Example 3.5 of
# Dimitrova, Kaishev, Tan (2020)

# When the sample size is larger than 100, the
# function dgof::ks.test will be numerically
# unstable

x3 <- sample(1:10, 25, replace = TRUE)
KSgeneral::disc_ks_test(x3, ecdf(1:10), exact = TRUE)
dgof::ks.test(x3, ecdf(1:10), exact = TRUE)
KSgeneral::disc_ks_test(x3, ecdf(1:10), exact = TRUE)$p -
          dgof::ks.test(x3, ecdf(1:10), exact = TRUE)$p

x4 <- sample(1:10, 500, replace = TRUE)
KSgeneral::disc_ks_test(x4, ecdf(1:10), exact = TRUE)
dgof::ks.test(x4, ecdf(1:10), exact = TRUE)
KSgeneral::disc_ks_test(x4, ecdf(1:10), exact = TRUE)$p -
          dgof::ks.test(x4, ecdf(1:10), exact = TRUE)$p

# Using stepfun() to specify the same discrete distribution as defined by ecdf():

steps <- stepfun(1:10, cumsum(c(0, rep(0.1, 10))))
KSgeneral::disc_ks_test(x3, steps, exact = TRUE)
```

`KS2sample` *Computes the p-value for a (weighted) two-sample Kolmogorov-Smirnov test, given an arbitrary positive weight function and arbitrary data samples with possibly repeated observations (i.e. ties)*

Description

Computes the p-value $P(D_{m,n} \geq q)$, where $D_{m,n}$ is the one- or two-sided two-sample Kolmogorov-Smirnov test statistic with weight function `weight`, when q = `d`, i.e. the observed value of KS statistic computed based on two data samples $\{x_1, ..., x_m\}$ and $\{y_1, ..., y_n\}$ that may come from continuous, discrete or mixed distribution, i.e. they may have repeated observations (ties).

Usage

```
KS2sample(x, y, alternative = c("two.sided", "less", "greater"),
conservative = F, weight = 0, tol = 1e-08, tail = T)
```

Arguments

`x`	a numeric vector of data sample values $\{x_1, ..., x_m\}$.
`y`	a numeric vector of data sample values $\{y_1, ..., y_n\}$
`alternative`	Indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater". One can specify just the initial letter of the string, but the argument name must be given in full, e.g. `alternative = "t"`. See 'Details' for the meaning of the possible values.
`conservative`	logical variable indicating whether ties should be considered. See 'Details' for the meaning.
`weight`	either a numeric value between 0 and 1 which specifies the form of the weight function from a class of pre-defined functions, or a user-defined strictly positive function of one variable. By default, no weight function is assumed. See 'Details' for the meaning of the possible values.
`tol`	the value of ϵ for computing $P(D_{m,n} > q-\epsilon)$, which is equivalent to $P(D_{m,n} \geq q)$. Non-positive input (`tol` ≤ 0) or large input (`tol` >`1e-6`) are replaced by `tol = 1e-6`. In cases when `m` and `n` have large least common multiple, a smaller value is highly recommended.
`tail`	logical variable indicating whether a p-value, $P(D_{m,n} \geq q)$ or one minus the p-value, $P(D_{m,n} < q)$, should be computed. By default, the p-value $P(D_{m,n} \geq q)$ is computed. See 'Details' for the meaning.

Details

Given a pair of random samples $\boldsymbol{X}_m = (X_1, ..., X_m)$ and $\boldsymbol{Y}_n = (Y_1, ..., Y_n)$ of sizes `m` and `n` with empirical cdfs $F_m(t)$ and $G_n(t)$ respectively, coming from some unknown cdfs $F(x)$ and $G(x)$. It is assumed that $F(x)$ and $G(x)$ could be either *continuous*, *discrete* or *mixed*, which means that repeated observations are allowed in the corresponding observed samples. The task is to test the null

hypothesis $H_0 : F(x) = G(x)$ for all x, either against the alternative hypothesis $H_1 : F(x) \neq G(x)$ for at least one x, which corresponds to the two-sided test, or against $H_1 : F(x) > G(x)$ and $H_1 : F(x) < G(x)$ for at least one x, which corresponds to the two one-sided tests. The (weighted) two-sample Kolmogorov-Smirnov goodness-of-fit statistics that are used to test these hypotheses are generally defined as:

$$\Delta_{m,n} = \sup |F_m(t) - G_n(t)| W(E_{m+n}(t), \text{ to test against the alternative } H_1 : F(x) \neq G(x)$$

$$\Delta^+_{m,n} = \sup [F_m(t) - G_n(x)] W(E_{m+n}(t)), \text{ to test against the alternative } H_1 : F(x) > G(x)$$

$$\Delta^-_{m,n} = \sup [G_n(t) - F_m(x)] W(E_{m+n}(t)), \text{ to test against the alternative } H_1 : F(x) < G(x),$$

where $E_{m+n}(t)$ is the empirical cdf of the pooled sample $\boldsymbol{Z}_{m,n} = (X_1, ..., X_m, Y_1, ..., Y_n)$, $W()$ is a strictly positive weight function defined on $[0, 1]$.

Possible values of `alternative` are `"two.sided"`, `"greater"` and `"less"` which specify the alternative hypothesis, i.e. specify the test statistics to be either $\Delta_{m,n}$, $\Delta^+_{m,n}$ or $\Delta^-_{m,n}$ respectively.

When `weight` is assigned with a numeric value ν between `0` and `1`, the test statistic is specified as the weighted two-sample Kolmorogov-Smirnov test with generalized Anderson-Darling weight $W(t) = 1/[t(1-t)]^\nu$ (see Finner and Gontscharuk 2018). Then for example, the two-sided two-sample Kolmogorov-Smirnov statistic has the following form:

$$\Delta_{m,n} = \sup_t \frac{|F_m(t) - G_n(t)|}{[E_{m+n}(t)(1 - E_{m+n}(t))]^\nu}$$

The latter specification defines a family of weighted Kolmogorov-Smirnov tests, covering the unweighted test (when `weight =` $\nu = 0$), and the widely-known weighted Kolmogorov-Smirnov test with Anderson-Darling weight (when `weight = 0.5`, see definition of this statistic also in Canner 1975). If one wants to implement a weighted test with a user-specified weight function, for example, $W(t) = 1/[t(2-t)]^{1/2}$ suggested by Buning (2001), which ensures higher power when both `x` and `y` come from distributions that are left-skewed and heavy-tailed, one can directly assign a univariate function with output value `1/sqrt(t*(2-t))` to `weight`. See 'Examples' for this demonstration.

For a particular realization of the pooled sample $\boldsymbol{Z}_{m,n}$, let there be k distinct values, $a_1 < a_2 < ... < a_k$, in the ordered, pooled sample $(z_1 \leq z_2 \leq \ldots \leq z_{m+n})$, where $k \leq m + n$, and where m_i is the number of times a_i, $i = 1, \ldots, k$ appears in the pooled sample. The p-value is then defined as the probability

$$p = P\left(D_{m,n} \geq q\right),$$

where $D_{m,n}$ is the two-sample Kolmogorov-Smirnov test statistic defined according to the value of `weight` and `alternative`, for two samples $\boldsymbol{X}'_m$ and $\boldsymbol{Y}'_n$ of sizes m and n, *randomly drawn from the pooled sample without replacement* and q = `d`, the observed value of the statistic calculated based on the user provided data samples `x` and `y`. By default `tail = T`, the p-value is returned, otherwise $1 - p$ is returned.

Note that, $D_{m,n}$ is defined on the space Ω of all possible pairs, $C = \frac{(m+n)!}{m!n!}$ of edfs $F_m(x, \omega)$ and $G_n(x, \omega)$, $\omega \in \Omega$, that correspond to the pairs of samples $\boldsymbol{X}'_m$ and $\boldsymbol{Y}'_n$, randomly drawn from, $\boldsymbol{Z}_{m+n}$, as follows. First, m observations are drawn at random without replacement, forming the first sample $\boldsymbol{X}'_m$, with corresponding edf, $F_m(x, \omega)$. The remaining n observations are then assigned to the second sample $\boldsymbol{Y}'_n$, with corresponding edf $G_n(x, \omega)$. Observations are then replaced back in $\boldsymbol{Z}_{m+n}$ and re-sampling is continued until the occurrence of all the C possible pairs of edfs $F_m(x, \omega)$ and $G_n(x, \omega)$, $\omega \in \Omega$. The pairs of edf's may be coincident if there are ties in the data and each pair, $F_m(x, \omega)$ and $G_n(x, \omega)$ occurs with probability $1/C$.

`conservative` is a logical variable whether the test should be conducted conservatively. By default, `conservative = F`, `KS2sample` returns the p-value that is defined through the conditional probability above. However, when the user has a priori knowledge that both samples are from a continuous distribution even if ties are present, for example, repeated observations are caused by rounding errors, the value `conservative = T` should be assigned, since the conditional probability is no longer relevant. In this case, `KS2sample` computes p-values for the Kolmogorov-Smirnov test assuming no ties are present, and returns a p-value which is an upper bound of the true p-value. Note that, if the null hypothesis is rejected using the calculated upper bound for the p-value, it should also be rejected with the true p-value.

`KS2sample` calculates the exact p-value of the KS test using an algorithm which generalizes the method due to Nikiforov (1994). If `tail = F`, `KS2sample` calculates the complementary p-value, $1-p$. For the purpose, an exact algorithm which generalizes the method due to Nikiforov (1994) is implemented. Alternatively, if `tail = T`, a version of the Nikiforov's recurrence proposed recently by Viehmann (2021) is implemented, which computes directly the p-value, with higher accuracy, giving up to 17 correct digits, but at up to 3 times higher computational cost. `KS2sample` ensures a total worst-case run-time of order $O(nm)$. In comparison with other known algorithms, it not only allows the flexible choice of weights which in some cases improve the statistical power (see Dimitrova, Jia, Kaishev 2024), but also is more efficient and generally applicable for *large sample sizes*.

Value

A list with class `"htest"` containing the following components:

`statistic`	the value of the test statistic `d`.
`p.value`	the p-value of the test.
`alternative`	a character string describing the alternative hypothesis.
`data.name`	a character string giving names of the data.

Source

Based on the Fortran subroutine by Nikiforov (1994). See also Dimitrova, Jia, Kaishev (2024).

References

Buning H (2001). "Kolmogorov-Smirnov- and Cramer-von Mises Type Two-sample Tests With Various Weight Functions." Communications in Statistics - Simulation and Computation, **30**(4), 847-865.

Finner H, Gontscharuk V (2018). "Two-sample Kolmogorov-Smirnov-type tests revisited: Old and new tests in terms of local levels." The Annals of Statistics, **46**(6A), 3014-3037.

Paul L. Canner (1975). "A Simulation Study of One- and Two-Sample Kolmogorov-Smirnov Statistics with a Particular Weight Function". Journal of the American Statistical Association, **70**(349), 209-211.

Nikiforov, A. M. (1994). "Algorithm AS 288: Exact Smirnov Two-Sample Tests for Arbitrary Distributions." Journal of the Royal Statistical Society. Series C (Applied Statistics), **43**(1), 265-270.

Viehmann, T. (2021). Numerically more stable computation of the p-values for the two-sample Kolmogorov-Smirnov test. *arXiv preprint* arXiv:2102.08037.

Dimitrina S. Dimitrova, Yun Jia, Vladimir K. Kaishev (2024). "The R functions KS2sample and Kuiper2sample: Efficient Exact Calculation of P-values of the Two-sample Kolmogorov-Smirnov and Kuiper Tests". *submitted*

Examples

```
##Computes p-value of two-sided unweighted test for continuous data
data1 <- rexp(750, 1)
data2 <- rexp(800, 1)
KS2sample(data1, data2)
##Computes the complementary p-value
KS2sample(data1, data2, tail = FALSE)
##Computes p-value of one-sided test with Anderson-Darling weight function
KS2sample(data1, data2, alternative = "greater", weight = 0.5)

##Computes p-values of two-sided test with Buning's weight function for discrete data
data3 <- rnbinom(100, size = 3, prob = 0.6)
data4 <- rpois(120, lambda = 2)
f <- function(t) 1 / sqrt( t * (2 - t) )
KS2sample(data3, data4, weight = f)
```

`KS2sample_c_Rcpp` *R function calling the C++ routines that compute the complementary p-value for a (weighted) two-sample Kolmogorov-Smirnov (KS) test, given an arbitrary positive weight function and arbitrary data samples with possibly repeated observations (i.e. ties)*

Description

Function calling directly the C++ routines that compute the exact complementary p-value $P(D_{m,n} < q)$ for the (weighed) two-sample one- or two-sided Kolmogorov-Smirnov statistic, at a fixed q, $q \in [0, 1]$, given the sample sizes `m` and `n`, the vector of weights `w_vec` and the vector `M` containing the number of times each distinct observation is repeated in the pooled sample.

Usage

```
KS2sample_c_Rcpp(m, n, kind, M, q, w_vec, tol)
```

Arguments

`m`	the sample size of first tested sample.
`n`	the sample size of second tested sample.
`kind`	an integer value (= 1,2 or 3) which specified the alternative hypothesis. When = 1, the test is two-sided. When = 2 or 3, the test is one-sided. See 'Details' for the meaning of the possible values. Other value is invalid.

M	an integer-valued vector with k cells, where k denotes the number of distinct values in the ordered pooled sample of tested pair of samples(i.e. $a_1 < a_2 < \ldots < a_k$). M[i] is the number of times that a_i is repeated in the pooled sample. A valid M must have strictly positive integer values and have the sum of all cells equals to m+n.
q	numeric value between 0 and 1, at which the p-value $P(D_{m,n} < q)$ is computed.
w_vec	a vector with m+n-1 cells, giving weights to each observation in the pooled sample. Valid w_vec must have m+n-1 cells and strictly positive value. See 'Details' for the meaning of values in each cell.
tol	the value of ϵ for computing $P(D_{m,n} \leq q-\epsilon)$, which is equivalent to $P(D_{m,n} < q)$. Non-positive input (tol ≤ 0) or large input (tol >1e-6) are replaced by tol=1e-6. In cases when m and n have large least common multiple, a smaller value is highly recommended.

Details

Given a pair of random samples $\boldsymbol{X}_m = (X_1, ..., X_m)$ and $\boldsymbol{Y}_n = (Y_1, ..., Y_n)$ of sizes m and n with empirical cdfs $F_m(t)$ and $G_n(t)$ respectively, coming from some unknown cdfs $F(x)$ and $G(x)$. It is assumed that $F(x)$ and $G(x)$ could be either *continuous*, *discrete* or *mixed*, which means that repeated observations are allowed in the corresponding observed samples. The task is to test the null hypothesis $H_0 : F(x) = G(x)$ for all x, either against the alternative hypothesis $H_1 : F(x) \neq G(x)$ for at least one x, which corresponds to the two-sided test, or against $H_1 : F(x) > G(x)$ and $H_1 : F(x) < G(x)$ for at least one x, which corresponds to the two one-sided tests. The (weighted) two-sample Kolmogorov-Smirnov goodness-of-fit statistics that are used to test these hypotheses are generally defined as:

$$\Delta_{m,n} = \sup |F_m(t) - G_n(t)| W(E_{m+n}(t), \text{ to test against the alternative } H_1 : F(x) \neq G(x)$$

$$\Delta^+_{m,n} = \sup [F_m(t) - G_n(x)] W(E_{m+n}(t)), \text{ to test against the alternative } H_1 : F(x) > G(x)$$

$$\Delta^-_{m,n} = \sup [G_n(t) - F_m(x)] W(E_{m+n}(t)), \text{ to test against the alternative } H_1 : F(x) < G(x),$$

where $E_{m+n}(t)$ is the empirical cdf of the pooled sample $\boldsymbol{Z}_{m,n} = (X_1, ..., X_m, Y_1, ..., Y_n)$, $W()$ is a strictly positive weight function defined on $[0, 1]$.

w_vec[i] (0<i<$m + n$) is then equal to $W(Z_i) = W(\frac{i}{m+n})$($Z_i$ is the i-th smallest observation in the pooled sample $\boldsymbol{Z}_{m,n}$). Different value of w_vec specifies the weighted Kolmogorov-Smirnov test differently. For example, when w_vec=rep(1,m+n-1), KS2sample_Rcpp calculates the p-value of the unweighted two-sample Kolmogorov-Smirnov test, when w_vec = ((1:(m+n-1))*((m+n-1):1))^(-1/2), it calculates the p-value for the weighted two-sample Kolmogorov-Smirnov test with Anderson-Darling weight $W(t) = 1/[t(1-t)]^{1/2}$.

Possible values of kind are 1,2 and 3, which specify the alternative hypothesis, i.e. specify the test statistic to be either $\Delta_{m,n}$, $\Delta^+_{m,n}$ or $\Delta^-_{m,n}$ respectively.

The numeric array M specifies the number of *repeated observations* in the pooled sample. For a particular realization of the pooled sample $\boldsymbol{Z}_{m,n} = (X_1, ..., X_m, Y_1, ..., Y_n)$, let there be k distinct values, $a_1 < a_2 < ... < a_k$, in the ordered, pooled sample ($z_1 \leq z_2 \leq \ldots \leq z_{m+n}$), where $k \leq m + n$, and where m_i=M[i] is the number of times a_i, $i = 1, \ldots, k$ appears in the pooled sample. The calculated complementary p-value is the conditional probability:

$$P(D_{m,n} < q)$$

where $D_{m,n}$ is the two-sample Kolmogorov-Smirnov test statistic defined according to the value of `weight` and `alternative`, for two samples $\boldsymbol{X}'_m$ and $\boldsymbol{Y}'_n$ of sizes m and n, *randomly drawn from the pooled sample without replacement*, i.e. $D_{m,n}$ is defined on the space Ω (see further details in `KS2sample`), and $q \in [0, 1]$.

`KS2sample_c_Rcpp` implements an exact algorithm, extending the Fortran 77 subroutine due to Nikiforov (1994), an extended functionality by allowing more flexible choice of weight, as well as for *large sample sizes*. This leads to faster computation time, as well as, relatively high accuracy for very large `m` and `n` (less accurate than `KS2sample_Rcpp`). Compared with other known algorithms, it allows data samples come from *continuous, discrete or mixed distribution*(i.e. ties may appear), and it is more efficient and more generally applicable for *large sample sizes*. This algorithm ensures a total worst-case run-time of order $O(nm)$.

Value

Numeric value corresponding to $P(D_{m,n} < q)$, given sample sizes `m`, `n`, `M` and `w_vec`. If the value of `m`, `n` are non-positive, or if the length of `w_vec` is not equal to `m+n-1`, then the function returns `-1`, the non-permitted value of `M` or non-permitted value inside `w_vec` returns `-2`, numerically unstable calculation returns `-3`.

Source

Based on the Fortran subroutine by Nikiforov (1994). See also Dimitrova, Jia, Kaishev (2024).

References

Paul L. Canner (1975). "A Simulation Study of One- and Two-Sample Kolmogorov-Smirnov Statistics with a Particular Weight Function". Journal of the American Statistical Association, **70**(349), 209-211.

Nikiforov, A. M. (1994). "Algorithm AS 288: Exact Smirnov Two-Sample Tests for Arbitrary Distributions." Journal of the Royal Statistical Society. Series C (Applied Statistics), **43**(1), 265–270.

Dimitrina S. Dimitrova, Yun Jia, Vladimir K. Kaishev (2024). "The R functions KS2sample and Kuiper2sample: Efficient Exact Calculation of P-values of the Two-sample Kolmogorov-Smirnov and Kuiper Tests". *submitted*

Examples

```
## Computing the unweighted two-sample Kolmogorov-Smirnov test
## Example see in Nikiforov (1994)

m <- 120
n <- 150
kind <- 1
q <- 0.1
M <- c(80,70,40,80)
w_vec <- rep(1,m+n-1)
tol <- 1e-6
KS2sample_c_Rcpp(m, n, kind, M, q, w_vec, tol)

kind <- 2
KS2sample_c_Rcpp(m, n, kind, M, q, w_vec, tol)
```

```
## Computing the weighted two-sample Kolmogorov-Smirnov test
## with Anderson-Darling weight
kind <- 3
w_vec <- ((1:(m+n-1))*((m+n-1):1))^(-1/2)
KS2sample_c_Rcpp(m, n, kind, M, q, w_vec, tol)
```

`KS2sample_Rcpp` *R function calling the C++ routines that compute the p-value for a (weighted) two-sample Kolmogorov-Smirnov (KS) test, given an arbitrary positive weight function and arbitrary data samples with possibly repeated observations (i.e. ties)*

Description

Function calling directly the C++ routines that compute the exact p-value $P(D_{m,n} \geq q)$ for the (weighed) two-sample one- or two-sided Kolmogorov-Smirnov statistic, at a fixed q, $q \in [0, 1]$, given the sample sizes `m` and `n`, the vector of weights `w_vec` and the vector `M` containing the number of times each distinct observation is repeated in the pooled sample.

Usage

```
KS2sample_Rcpp(m, n, kind, M, q, w_vec, tol)
```

Arguments

`m`	the sample size of first tested sample.
`n`	the sample size of second tested sample.
`kind`	an integer value (= 1,2 or 3) which specified the alternative hypothesis. When = 1, the test is two-sided. When = 2 or 3, the test is one-sided. See 'Details' for the meaning of the possible values. Other value is invalid.
`M`	an integer-valued vector with k cells, where k denotes the number of distinct values in the ordered pooled sample of tested pair of samples(i.e. $a_1 < a_2 < \ldots < a_k$). `M[i]` is the number of times that a_i is repeated in the pooled sample. A valid `M` must have strictly positive integer values and have the sum of all cells equals to `m+n`.
`q`	numeric value between 0 and 1, at which the p-value $P(D_{m,n} \geq q)$ is computed.
`w_vec`	a vector with `m+n-1` cells, giving weights to each observation in the pooled sample. Valid `w_vec` must have `m+n-1` cells and strictly positive value. See 'Details' for the meaning of values in each cell.
`tol`	the value of ϵ for computing $P(D_{m,n} > q - \epsilon)$, which is equivalent to $P(D_{m,n} \geq q)$. Non-positive input (`tol` ≤ 0) or large input (`tol` >`1e-6`) are replaced by `tol` = `1e-6`. In cases when `m` and `n` have large least common multiple, a smaller value is highly recommended.

Details

Given a pair of random samples $\boldsymbol{X}_m = (X_1, ..., X_m)$ and $\boldsymbol{Y}_n = (Y_1, ..., Y_n)$ of sizes `m` and `n` with empirical cdfs $F_m(t)$ and $G_n(t)$ respectively, coming from some unknown cdfs $F(x)$ and $G(x)$. It is assumed that $F(x)$ and $G(x)$ could be either *continuous*, *discrete* or *mixed*, which means that repeated observations are allowed in the corresponding observed samples. The task is to test the null hypothesis $H_0 : F(x) = G(x)$ for all x, either against the alternative hypothesis $H_1 : F(x) \neq G(x)$ for at least one x, which corresponds to the two-sided test, or against $H_1 : F(x) > G(x)$ and $H_1 : F(x) < G(x)$ for at least one x, which corresponds to the two one-sided tests. The (weighted) two-sample Kolmogorov-Smirnov goodness-of-fit statistics that are used to test these hypotheses are generally defined as:

$$\Delta_{m,n} = \sup |F_m(t) - G_n(t)| W(E_{m+n}(t), \text{ to test against the alternative } H_1 : F(x) \neq G(x)$$

$$\Delta^+_{m,n} = \sup [F_m(t) - G_n(x)] W(E_{m+n}(t)), \text{ to test against the alternative } H_1 : F(x) > G(x)$$

$$\Delta^-_{m,n} = \sup [G_n(t) - F_m(x)] W(E_{m+n}(t)), \text{ to test against the alternative } H_1 : F(x) < G(x),$$

where $E_{m+n}(t)$ is the empirical cdf of the pooled sample $\boldsymbol{Z}_{m,n} = (X_1, ..., X_m, Y_1, ..., Y_n)$, $W()$ is a strictly positive weight function defined on $[0, 1]$.

`w_vec[i]` (0<`i`<$m + n$) is then equal to $W(Z_i) = W(\frac{i}{m+n})$($Z_i$ is the i-th smallest observation in the pooled sample $\boldsymbol{Z}_{m,n}$). Different value of `w_vec` specifies the weighted Kolmogorov-Smirnov test differently. For example, when `w_vec=rep(1,m+n-1)`, `KS2sample_Rcpp` calculates the p-value of the unweighted two-sample Kolmogorov-Smirnov test, when `w_vec = ((1:(m+n-1))*((m+n-1):1))^(-1/2)`, it calculates the p-value for the weighted two-sample Kolmogorov-Smirnov test with Anderson-Darling weight $W(t) = 1/[t(1-t)]^{1/2}$.

Possible values of `kind` are 1,2 and 3, which specify the alternative hypothesis, i.e. specify the test statistic to be either $\Delta_{m,n}$, $\Delta^+_{m,n}$ or $\Delta^-_{m,n}$ respectively.

The numeric array `M` specifies the number of *repeated observations* in the pooled sample. For a particular realization of the pooled sample $\boldsymbol{Z}_{m,n} = (X_1, ..., X_m, Y_1, ..., Y_n)$, let there be k distinct values, $a_1 < a_2 < ... < a_k$, in the ordered, pooled sample $(z_1 \leq z_2 \leq \ldots \leq z_{m+n})$, where $k \leq m + n$, and where m_i=`M[i]` is the number of times a_i, $i = 1, \ldots, k$ appears in the pooled sample. The p-value is then defined as the probability

$$P\left(D_{m,n} \geq q\right),$$

where $D_{m,n}$ is the two-sample Kolmogorov-Smirnov test statistic defined according to the value of `weight` and `alternative`, for two samples $\boldsymbol{X}'_m$ and $\boldsymbol{Y}'_n$ of sizes m and n, *randomly drawn from the pooled sample without replacement*, i.e. $D_{m,n}$ is defined on the space Ω (see further details in `KS2sample`), and $q \in [0, 1]$.

`KS2sample_Rcpp` implements an exact algorithm, extending the Fortran 77 subroutine due to Nikiforov (1994), an extended functionality by allowing more flexible choices of weight, as well as for *large sample sizes*. A version of the Nikiforov's recurrence proposed recently by Viehmann (2021) is further incorporated, which computes directly the p-value, with higher accuracy, giving up to 17 correct digits, but at up to 3 times higher computational cost than `KS2sample_c_Rcpp`. Compared with other known algorithms, it allows data samples to come from *continuous, discrete or mixed distribution*(i.e. ties may appear), and it is more efficient and more generally applicable for *large sample sizes*. This algorithm ensures a total worst-case run-time of order $O(nm)$.

Value

Numeric value corresponding to $P(D_{m,n} \geq q)$, given sample sizes `m`, `n`, `M` and `w_vec`. If the value of `m`, `n` are non-positive, or if the length of `w_vec` is not equal to `m+n-1`, then the function returns `-1`, the non-permitted value of `M` or non-permitted value inside `w_vec` returns `-2`, numerically unstable calculation returns `-3`.

Source

Based on the Fortran subroutine by Nikiforov (1994). See also Dimitrova, Jia, Kaishev (2024).

References

Paul L. Canner (1975). "A Simulation Study of One- and Two-Sample Kolmogorov-Smirnov Statistics with a Particular Weight Function". Journal of the American Statistical Association, **70**(349), 209-211.

Nikiforov, A. M. (1994). "Algorithm AS 288: Exact Smirnov Two-Sample Tests for Arbitrary Distributions." Journal of the Royal Statistical Society. Series C (Applied Statistics), **43**(1), 265–270.

Viehmann, T. (2021). Numerically more stable computation of the p-values for the two-sample Kolmogorov-Smirnov test. *arXiv preprint* arXiv:2102.08037.

Dimitrina S. Dimitrova, Yun Jia, Vladimir K. Kaishev (2024). "The R functions KS2sample and Kuiper2sample: Efficient Exact Calculation of P-values of the Two-sample Kolmogorov-Smirnov and Kuiper Tests". *submitted*

Examples

```
## Computing the unweighted two-sample Kolmogorov-Smirnov test
## Example see in Nikiforov (1994)

m <- 120
n <- 150
kind <- 1
q <- 0.1
M <- c(80,70,40,80)
w_vec <- rep(1,m+n-1)
tol <- 1e-6
KS2sample_Rcpp(m, n, kind, M, q, w_vec, tol)

kind <- 2
KS2sample_Rcpp(m, n, kind, M, q, w_vec, tol)

## Computing the weighted two-sample Kolmogorov-Smirnov test
## with Anderson-Darling weight
kind <- 3
w_vec <- ((1:(m+n-1))*((m+n-1):1))^(-1/2)
KS2sample_Rcpp(m, n, kind, M, q, w_vec, tol)
```

ks_c_cdf_Rcpp *R function calling directly the C++ routines that compute the complementary cumulative distribution function of the two-sided (or one-sided, as a special case) Kolmogorov-Smirnov statistic, when the cdf under the null hypothesis is arbitrary (i.e., purely discrete, mixed or continuous)*

Description

Function calling directly the C++ routines that compute the complementary cdf for the one-sample two-sided Kolmogorov-Smirnov statistic, given the sample size n and the file "Boundary_Crossing_Time.txt" in the working directory. The latter file contains A_i and B_i, $i = 1, ..., n$, specified in Steps 1 and 2 of the Exact-KS-FFT method (see Equation (5) in Section 2 of Dimitrova, Kaishev, Tan (2020)). The latter values form the n-dimensional rectangular region for the uniform order statistics (see Equations (3), (5) and (6) in Dimitrova, Kaishev, Tan (2020)), namely $P(D_n \geq q) = 1 - P(A_i \leq U_{(i)} \leq B_i, 1 \leq i \leq n) = 1 - P(g(t) \leq nU_n(t) \leq h(t), 0 \leq t \leq 1)$, where the upper and lower boundary functions $h(t)$, $g(t)$ are defined as $h(t) = \sum_{i=1}^{n} 1_{(A_i<t)}$, $g(t) = \sum_{i=1}^{n} 1_{(B_i \leq t)}$, or equivalently, noting that $h(t)$ and $g(t)$ are correspondingly left and right continuous functions, we have $\sup\{t \in [0,1] : h(t) < i\} = A_i$ and $\inf\{t \in [0,1] : g(t) > i-1\} = B_i$.

Note that on can also compute the (complementary) cdf for the one-sided KS statistics D_n^- or D_n^+ (cf., Dimitrova, Kaishev, Tan (2020)) by appropriately specifying correspondingly $A_i = 0$ for all i or $B_i = 1$ for all i, in the function ks_c_cdf_Rcpp.

Usage

```
ks_c_cdf_Rcpp(n)
```

Arguments

n	the sample size

Details

Note that all calculations here are done directly in C++ and output in R. That leads to faster computation time, as well as in some cases, possibly higher accuracy (depending on the accuracy of the pre-computed values A_i and B_i, $i = 1, ..., n$, provided in the file "Boundary_Crossing_Time.txt") compared to the functions cont_ks_c_cdf, disc_ks_c_cdf, mixed_ks_c_cdf.

Given a random sample $\{X_1, ..., X_n\}$ of size n with an empirical cdf $F_n(x)$, the two-sided Kolmogorov-Smirnov goodness-of-fit statistic is defined as $D_n = \sup |F_n(x) - F(x)|$, where $F(x)$ is the cdf of a prespecified theoretical distribution under the null hypothesis H_0, that $\{X_1, ..., X_n\}$ comes from $F(x)$. The one-sided KS test statistics are correspondingly defined as $D_n^- = \sup_x(F(x) - F_n(x))$ and $D_n^+ = \sup_x(F_n(x) - F(x))$.

The function ks_c_cdf_Rcpp implements the Exact-KS-FFT method, proposed by Dimitrova, Kaishev, Tan (2020), to compute the complementary cdf, $P(D_n \geq q)$ at a value q, when $F(x)$ is arbitrary (i.e. purely discrete, mixed or continuous). It is based on expressing the complementary

cdf as $P(D_n \geq q) = 1 - P(A_i \leq U_{(i)} \leq B_i, 1 \leq i \leq n)$, where A_i and B_i are defined as in Step 1 of Dimitrova, Kaishev, Tan (2020).

The complementary cdf is then re-expressed in terms of the conditional probability that a homogeneous Poisson process, $\xi_n(t)$ with intensity n will not cross an upper boundary $h(t)$ and a lower boundary $g(t)$, given that $\xi_n(1) = n$ (see Steps 2 and 3 in Section 2.1 of Dimitrova, Kaishev, Tan (2020)). This conditional probability is evaluated using FFT in Step 4 of the method in order to obtain the value of the complementary cdf $P(D_n \geq q)$. This algorithm ensures a total worst-case run-time of order $O(n^2 log(n))$ which makes it highly computationally efficient compared to other known algorithms developed for the special cases of continuous or purely discrete $F(x)$.

The values A_i and B_i, $i = 1, ..., n$, specified in Steps 1 and 2 of the Exact-KS-FFT method (see Dimitrova, Kaishev, Tan (2020), Section 2) must be pre-computed (in R or, if needed, using alternative softwares offering high accuracy, e.g. Mathematica) and saved in a file with the name "Boundary_Crossing_Time.txt" (in the current working directory).

The function `ks_c_cdf_Rcpp` is called in R and it first reads the file "Boundary_Crossing_Time.txt" and then computes the value for the complementaty cdf $P(D_n \geq q) = 1 - P(A_i \leq U_{(i)} \leq B_i, 1 \leq i \leq n) = 1 - P(g(t) \leq nU_n(t) \leq h(t), 0 \leq t \leq 1)$ in C++ and output in R (or as noted above, as a special case, computes the value of the complementary cdf $P(D_n^+ \geq q) = 1 - P(A_i \leq U_{(i)} \leq 1, 1 \leq i \leq n)$ or $P(D_n^- \geq q) = 1 - P(0 \leq U_{(i)} \leq B_i, 1 \leq i \leq n)$).

Value

Numeric value corresponding to $P(D_n \geq q) = 1 - P(A_i \leq U_{(i)} \leq B_i, 1 \leq i \leq n) = 1 - P(g(t) \leq \eta_n(t) \leq h(t), 0 \leq t \leq 1)$ (or, as a special case, to $P(D_n^+ \geq q)$ or $P(D_n^- \geq q)$), given a sample size n and the file "Boundary_Crossing_Time.txt" containing A_i and B_i, $i = 1, ..., n$, specified in Steps 1 and 2 of the Exact-KS-FFT method (see Dimitrova, Kaishev, Tan (2020), Section 2).

References

Dimitrina S. Dimitrova, Vladimir K. Kaishev, Senren Tan. (2020) "Computing the Kolmogorov-Smirnov Distribution When the Underlying CDF is Purely Discrete, Mixed or Continuous". Journal of Statistical Software, **95**(10): 1-42. doi:10.18637/jss.v095.i10.

Moscovich A., Nadler B. (2017). "Fast Calculation of Boundary Crossing Probabilities for Poisson Processes". Statistics and Probability Letters, **123**, 177-182.

Examples

```
## Computing the complementary cdf P(D_{n} >= q)
## for n = 10 and q = 0.1, when F(x) is continuous,
## In this case,
## B_i = (i-1)/n + q
## A_i =  i/n - q


n <- 10
q <- 0.1
up_rec <- ((1:n)-1)/n + q
low_rec <- (1:n)/n - q
df <- data.frame(rbind(up_rec, low_rec))
write.table(df,"Boundary_Crossing_Time.txt", sep = ", ",
```

```
                    row.names = FALSE, col.names = FALSE)
ks_c_cdf_Rcpp(n)
```

`Kuiper2sample` *Computes the p-value for a two-sample Kuiper test, given arbitrary data samples on the real line or on the circle with possibly repeated observations (i.e. ties)*

Description

Computes the p-value, $P(V_{m,n} \geq q)$, where $V_{m,n}$ is the two-sample Kuiper test statistic, q = v, i.e. the observed value of the Kuiper statistic, computed based on two data samples $\{x_1, ..., x_m\}$ and $\{y_1, ..., y_n\}$ that may come from continuous, discrete or mixed distribution, i.e. they may have repeated observations (ties).

Usage

```
Kuiper2sample(x, y, conservative = F, tail = T)
```

Arguments

`x`	a numeric vector of data sample values $\{x_1, ..., x_m\}$
`y`	a numeric vector of data sample values $\{y_1, ..., y_n\}$
`conservative`	logical variable indicating whether ties should be considered. See 'Details' for the meaning.
`tail`	logical variable indicating whether a p-value, $P(V_{m,n} \geq q)$ or one minus the p-value, $P(V_{m,n} < q)$, should be computed. By default, the p-value $P(V_{m,n} \geq q)$ is computed. See 'Details' for the meaning.

Details

Given a pair of random samples, either on the real line or the circle, denoted by $\boldsymbol{X}_m = (X_1, ..., X_m)$ and $\boldsymbol{Y}_n = (Y_1, ..., Y_n)$, of sizes `m` and `n` with empirical cdfs $F_m(t)$ and $G_n(t)$ respectively, coming from some unknown cdfs $F(x)$ and $G(x)$. It is assumed that $F(x)$ and $G(x)$ could be either *continuous*, *discrete* or *mixed*, which means that repeated observations are allowed in the corresponding observed samples. The task is to test the null hypothesis $H_0 : F(x) = G(x)$ for all x, against the alternative hypothesis $H_1 : F(x) \neq G(x)$ for at least one x. The two-sample Kuiper goodness-of-fit statistic that is used to test this hypothesis is defined as:

$$\varsigma_{m,n} = \sup[F_m(t) - G_n(t)] - \inf[F_m(t) - G_n(t)].$$

For a particular realization of the pooled sample $\boldsymbol{Z}_{m,n} = (X_1, ..., X_m, Y_1, ..., Y_n)$, let there be k distinct values, $a_1 < a_2 < ... < a_k$, in the ordered, pooled sample $(z_1 \leq z_2 \leq \ldots \leq z_{m+n})$, where $k \leq m + n$, and where m_i is the number of times a_i, $i = 1, \ldots, k$ appears in the pooled sample. The p-value is then defined as the probability

$$p = P\left(V_{m,n} \geq q\right),$$

where $V_{m,n}$ is the two-sample Kuiper test statistic defined as $\varsigma_{m,n}$, for two samples $\boldsymbol{X}'_m$ and $\boldsymbol{Y}'_n$ of sizes m and n, *randomly drawn from the pooled sample without replacement* and q = v, the observed value of the statistic calculated based on the user provided data samples x and y. By default tail = T, the p-value is returned, otherwise $1-p$ is returned.

Note that, $V_{m,n}$ is defined on the space Ω of all possible pairs, $C = \frac{(m+n)!}{m!n!}$ of edfs $F_m(x,\omega)$ and $G_n(x,\omega)$, $\omega \in \Omega$, that correspond to the pairs of samples $\boldsymbol{X}'_m$ and $\boldsymbol{Y}'_n$, randomly drawn from, $\boldsymbol{Z}_{m+n}$, as follows. First, m observations are drawn at random without replacement, forming the first sample $\boldsymbol{X}'_m$, with corresponding edf, $F_m(x,\omega)$. The remaining n observations are then assigned to the second sample $\boldsymbol{Y}'_n$, with corresponding edf $G_n(x,\omega)$. Observations are then replaced back in $\boldsymbol{Z}_{m+n}$ and re-sampling is continued until the occurrence of all the C possible pairs of edfs $F_m(x,\omega)$ and $G_n(x,\omega)$, $\omega \in \Omega$. The pairs of edf's may be coincident if there are ties in the data and each pair, $F_m(x,\omega)$ and $G_n(x,\omega)$ occurs with probability $1/C$.

conservative is a logical variable whether the test should be conducted conservatively. By default, conservative = F, Kuiper2sample returns the p-value that is defined through the conditional probability above. However, when the user has a priori knowledge that both samples are from a continuous distribution even if ties are present, for example, repeated observations are caused by rounding errors, the value conservative = T should be assigned, since the conditional probability is no longer relevant. In this case, Kuiper2sample computes p-values for the Kuiper test assuming no ties are present, and returns a p-value which is an upper bound of the true p-value. Note that, if the null hypothesis is rejected using the calculated upper bound for the p-value, it should also be rejected with the true p-value.

Kuiper2sample calculates the exact p-value of the Kuiper test using an algorithm from Dimitrova, Jia, Kaishev (2024), which is based on extending the algorithm provided by Nikiforov (1994) and generalizing the method due to Maag and Stephens (1968) and Hirakawa (1973). If tail = F, Kuiper2sample calculates the complementary p-value $1-p$. For the purpose, an exact algorithm which generalizes the method due to Nikiforov (1994) is implemented. Alternatively, if tail = T, a version of the Nikiforov's recurrence proposed recently by Viehmann (2021) is further incorporated, which computes directly the p-value, with up to 4 digits extra accuracy, but at up to 3 times higher computational cost. It is accurate and valid for *arbitrary (possibly large) sample sizes*. This algorithm ensures a total worst-case run-time of order $O((mn)^2)$. When m and n have large greatest common divisor (an extreme case is m = n), it ensures a total worst-case run-time of order $O((m)^2 n)$.

Kuiper2sample is accurate and fast compared with the function based on the Monte Carlo simulation. Compared to the implementation using asymptotic method, Kuiper2sample allows data samples to come from *continuous, discrete or mixed distribution* (i.e. ties may appear), and is more accurate than asymptotic method when sample sizes are small.

Value

A list with class "htest" containing the following components:

statistic	the value of the test statistic v.
p.value	the p-value of the test.
alternative	a character string describing the alternative hypothesis.
data.name	a character string giving names of the data.

References

Maag, U. R., Stephens, M. A. (1968). The V_{NM} Two-Sample Test. The Annals of Mathematical Statistics, **39**(3), 923-935.

Hirakawa, K. (1973). The two-sample Kuiper test. TRU Mathematics, **9**, 99-118.

Nikiforov, A. M. (1994). "Algorithm AS 288: Exact Smirnov Two-Sample Tests for Arbitrary Distributions." Journal of the Royal Statistical Society. Series C (Applied Statistics), **43**(1), 265–270.

Viehmann, T. (2021). Numerically more stable computation of the p-values for the two-sample Kolmogorov-Smirnov test. *arXiv preprint* arXiv:2102.08037.

Dimitrina S. Dimitrova, Yun Jia, Vladimir K. Kaishev (2024). "The R functions KS2sample and Kuiper2sample: Efficient Exact Calculation of P-values of the Two-sample Kolmogorov-Smirnov and Kuiper Tests". *submitted*

Examples

```
##Computes discrete circular data
data1 <- c(rep(pi/2,30),rep(pi,30),rep(3*pi/2,30),rep(2*pi,30))
data2 <- c(rep(pi/2,50),rep(pi,40),rep(3*pi/2,10),rep(2*pi,50))
Kuiper2sample(data1, data2)

##The calculated p-value does not change with the choice of the original point
data3 <- c(rep(pi/2,30),rep(pi,30),rep(3*pi/2,30),rep(2*pi,30))
data4 <- c(rep(pi/2,50),rep(pi,50),rep(3*pi/2,40),rep(2*pi,10))
Kuiper2sample(data3, data4)
```

Kuiper2sample_c_Rcpp

R function calling the C++ routines that compute the complementary p-value for a (unweighted) two-sample Kuiper test, given arbitrary data samples on the real line or on the circle with possibly repeated observations (i.e. ties)

Description

Function calling directly the C++ routines that compute the exact complementary p-value $P(V_{m,n} < q)$ for the two-sample Kuiper test, at a fixed q, $q \in [0, 2]$, given the sample sizes `m`, `n` and the vector `M` containing the number of times each distinct observation is repeated in the pooled sample.

Usage

```
Kuiper2sample_c_Rcpp(m, n, M, q)
```

Arguments

`m`	the sample size of first tested sample.
`n`	the sample size of second tested sample.

M — an integer-valued vector with k cells, where k denotes the number of distinct values in the ordered pooled sample of tested pair of samples(i.e. $a_1 < a_2 < \ldots < a_k$). M[i] is the number of times that a_i is repeated in the pooled sample. A valid M must have strictly positive integer values and have the sum of all cells equals to m+n.

q — numeric value between 0 and 2, at which the p-value $P(V_{m,n} < q)$ is computed.

Details

Given a pair of random samples, either on the real line or the circle, denoted by $\boldsymbol{X}_m = (X_1, ..., X_m)$ and $\boldsymbol{Y}_n = (Y_1, ..., Y_n)$, of sizes m and n with empirical cdfs $F_m(t)$ and $G_n(t)$ respectively, coming from some unknown cdfs $F(x)$ and $G(x)$. It is assumed that $F(x)$ and $G(x)$ could be either *continuous*, *discrete* or *mixed*, which means that repeated observations are allowed in the corresponding observed samples. The task is to test the null hypothesis $H_0 : F(x) = G(x)$ for all x, against the alternative hypothesis $H_1 : F(x) \neq G(x)$ for at least one x. The two-sample Kuiper goodness-of-fit statistic that is used to test this hypothesis is defined as:

$$\varsigma_{m,n} = \sup[F_m(t) - G_n(t)] - \inf[F_m(t) - G_n(t)].$$

The numeric array M specifies the number of *repeated observations* in the pooled sample. For a particular realization of the pooled sample $\boldsymbol{Z}_{m,n} = (X_1, ..., X_m, Y_1, ..., Y_n)$, let there be k distinct values, $a_1 < a_2 < ... < a_k$, in the ordered, pooled sample $(z_1 \leq z_2 \leq \ldots \leq z_{m+n})$, where $k \leq m + n$, and where m_i = M[i] is the number of times a_i, $i = 1, \ldots, k$ appears in the pooled sample. The calculated complementary p-value is then the conditional probability:

$$P(V_{m,n} < q)$$

where $V_{m,n}$ is the two-sample Kuiper test statistic defined as $\varsigma_{m,n}$, for two samples $\boldsymbol{X}'_m$ and $\boldsymbol{Y}'_n$ of sizes m and n, *randomly drawn from the pooled sample without replacement*, i.e. $V_{m,n}$ is defined on the space Ω (see further details in Kuiper2sample), and $q \in [0, 2]$.

Kuiper2sample_c_Rcpp implements an algorithm from Dimitrova, Jia, Kaishev (2024), that is based on extending the algorithm provided by Nikiforov (1994) and generalizing the method due to Maag and Stephens (1968) and Hirakawa (1973). It is relatively accurate (less accurate than Kuiper2sample_Rcpp) and valid for *arbitrary (possibly large) sample sizes*. This algorithm ensures a total worst-case run-time of order $O((mn)^2)$. When m and n have large greatest common divisor (an extreme case is m = n), it ensures a total worst-case run-time of order $O((m)^2 n)$.

Other known implementations for the two-sample Kuiper test mainly use the approximation method or Monte Carlo simulation (See also Kuiper2sample). The former method is invalid for data with ties and often gives p-values with large errors when sample sizes are small, the latter method is usually slow and inaccurate. Compared with other known algorithms, Kuiper2sample_c_Rcpp allows data samples to come from *continuous, discrete or mixed distribution* (i.e. ties may appear), and is more accurate and generally applicable for *large sample sizes*.

Value

Numeric value corresponding to $P(V_{m,n} < q)$, given sample sizes m, n and M. If the value of m, n are non-positive, or their least common multiple exceeds the limit 2147483647, then the function returns -1, the non-permitted value of M returns -2, numerically unstable calculation returns -3.

References

Maag, U. R., Stephens, M. A. (1968). The V_{NM} Two-Sample Test. The Annals of Mathematical Statistics, **39**(3), 923-935.

Hirakawa, K. (1973). The two-sample Kuiper test. TRU Mathematics, **9**, 99-118.

Nikiforov, A. M. (1994). "Algorithm AS 288: Exact Smirnov Two-Sample Tests for Arbitrary Distributions." Journal of the Royal Statistical Society. Series C (Applied Statistics), **43**(1), 265–270.

Dimitrina S. Dimitrova, Yun Jia, Vladimir K. Kaishev (2024). "The R functions KS2sample and Kuiper2sample: Efficient Exact Calculation of P-values of the Two-sample Kolmogorov-Smirnov and Kuiper Tests". *submitted*

Examples

```
## Computing the unweighted two-sample Kolmogorov-Smirnov test
## Example see in Nikiforov (1994)

m <- 120
n <- 150
q <- 0.183333333
M <- c(80,70,40,80)
Kuiper2sample_c_Rcpp(m, n, M, q)
```

Kuiper2sample_Rcpp — *R function calling the C++ routines that compute the p-value for a (unweighted) two-sample Kuiper test, given arbitrary data samples on the real line or on the circle with possibly repeated observations (i.e. ties)*

Description

Function calling directly the C++ routines that compute the exact p-value $P(V_{m,n} \geq q)$ for the two-sample Kuiper test, at a fixed q, $q \in [0, 2]$, given the sample sizes `m`, `n` and the vector `M` containing the number of times each distinct observation is repeated in the pooled sample.

Usage

```
Kuiper2sample_Rcpp(m, n, M, q)
```

Arguments

`m`	the sample size of first tested sample.
`n`	the sample size of second tested sample.
`M`	an integer-valued vector with k cells, where k denotes the number of distinct values in the ordered pooled sample of tested pair of samples(i.e. $a_1 < a_2 < \ldots < a_k$). `M[i]` is the number of times that a_i is repeated in the pooled sample. A valid `M` must have strictly positive integer values and have the sum of all cells equals to `m+n`.
`q`	numeric value between 0 and 2, at which the p-value $P(V_{m,n} \geq q)$ is computed.

Details

Given a pair of random samples, either on the real line or the circle, denoted by $\boldsymbol{X}_m = (X_1, ..., X_m)$ and $\boldsymbol{Y}_n = (Y_1, ..., Y_n)$, of sizes `m` and `n` with empirical cdfs $F_m(t)$ and $G_n(t)$ respectively, coming from some unknown cdfs $F(x)$ and $G(x)$. It is assumed that $F(x)$ and $G(x)$ could be either *continuous*, *discrete* or *mixed*, which means that repeated observations are allowed in the corresponding observed samples. The task is to test the null hypothesis $H_0 : F(x) = G(x)$ for all x, against the alternative hypothesis $H_1 : F(x) \neq G(x)$ for at least one x. The two-sample Kuiper goodness-of-fit statistic that is used to test this hypothesis is defined as:

$$\varsigma_{m,n} = \sup[F_m(t) - G_n(t)] - \inf[F_m(t) - G_n(t)].$$

The numeric array `M` specifies the number of *repeated observations* in the pooled sample. For a particular realization of the pooled sample $\boldsymbol{Z}_{m,n} = (X_1, ..., X_m, Y_1, ..., Y_n)$, let there be k distinct values, $a_1 < a_2 < ... < a_k$, in the ordered, pooled sample $(z_1 \leq z_2 \leq \ldots \leq z_{m+n})$, where $k \leq m + n$, and where m_i = `M[i]` is the number of times a_i, $i = 1, \ldots, k$ appears in the pooled sample. The p-value is then defined as the probability

$$P\left(V_{m,n} \geq q\right),$$

where $V_{m,n}$ is the two-sample Kuiper test statistic defined as $\varsigma_{m,n}$, for two samples $\boldsymbol{X}'_m$ and $\boldsymbol{Y}'_n$ of sizes m and n, *randomly drawn from the pooled sample without replacement*, i.e. $V_{m,n}$ is defined on the space Ω (see further details in `Kuiper2sample`), and $q \in [0, 2]$.

`Kuiper2sample_Rcpp` implements an algorithm from Dimitrova, Jia, Kaishev (2024), that is based on extending the algorithm provided by Nikiforov (1994) and generalizing the method due to Maag and Stephens (1968) and Hirakawa (1973). A version of the Nikiforov's recurrence proposed recently by Viehmann (2021) is further incorporated, which computes directly the p-value, with up to 4 digits extra accuracy, but at up to 3 times higher computational cost than `Kuiper2sample_c_Rcpp`. It is accurate and valid for *arbitrary (possibly large) sample sizes*. This algorithm ensures a total worst-case run-time of order $O((mn)^2)$. When `m` and `n` have large greatest common divisor (an extreme case is `m = n`), it ensures a total worst-case run-time of order $O((m)^2 n)$.

Other known implementations for the two-sample Kuiper test mainly use the approximation method or Monte Carlo simulation (See also `Kuiper2sample`). The former method is invalid for data with ties and often gives p-values with large errors when sample sizes are small, the latter method is usually slow and inaccurate. Compared with other known algorithms, `Kuiper2sample_Rcpp` allows data samples to come from *continuous, discrete or mixed distribution* (i.e. ties may appear), and is more accurate and generally applicable for *large sample sizes*.

Value

Numeric value corresponding to $P(V_{m,n} \geq q)$, given sample sizes `m`, `n` and `M`. If the value of `m`, `n` are non-positive, or their least common multiple exceeds the limit 2147483647, then the function returns `-1`, the non-permitted value of `M` returns `-2`, numerically unstable calculation returns `-3`.

References

Maag, U. R., Stephens, M. A. (1968). The V_{NM} Two-Sample Test. The Annals of Mathematical Statistics, **39**(3), 923-935.

Hirakawa, K. (1973). The two-sample Kuiper test. TRU Mathematics, **9**, 99-118.

Nikiforov, A. M. (1994). "Algorithm AS 288: Exact Smirnov Two-Sample Tests for Arbitrary Distributions." Journal of the Royal Statistical Society. Series C (Applied Statistics), **43**(1), 265–270.

Viehmann, T. (2021). Numerically more stable computation of the p-values for the two-sample Kolmogorov-Smirnov test. *arXiv preprint* arXiv:2102.08037.

Dimitrina S. Dimitrova, Yun Jia, Vladimir K. Kaishev (2024). "The R functions KS2sample and Kuiper2sample: Efficient Exact Calculation of P-values of the Two-sample Kolmogorov-Smirnov and Kuiper Tests". *submitted*

Examples

```
## Computing the unweighted two-sample Kolmogorov-Smirnov test
## Example see in Nikiforov (1994)

m <- 120
n <- 150
q <- 0.183333333
M <- c(80,70,40,80)
Kuiper2sample_Rcpp(m, n, M, q)
```

`mixed_ks_c_cdf` *Computes the complementary cumulative distribution function of the two-sided Kolmogorov-Smirnov statistic when the cdf under the null hypothesis is mixed*

Description

Computes the complementary cdf, $P(D_n \geq q)$ at a fixed q, $q \in [0,1]$, of the one-sample two-sided Kolmogorov-Smirnov statistic, when the cdf $F(x)$ under the null hypothesis is mixed, using the Exact-KS-FFT method expressing the p-value as a double-boundary non-crossing probability for a homogeneous Poisson process, which is then efficiently computed using FFT (see Dimitrova, Kaishev, Tan (2020)).

Usage

```
mixed_ks_c_cdf(q, n, jump_points, Mixed_dist, ..., tol = 1e-10)
```

Arguments

`q`	numeric value between 0 and 1, at which the complementary cdf $P(D_n \geq q)$ is computed
`n`	the sample size
`jump_points`	a numeric vector containing the points of (jump) discontinuity, i.e. where the underlying cdf $F(x)$ has jump(s)
`Mixed_dist`	a pre-specified (user-defined) mixed cdf, $F(x)$, under the null hypothesis.
`...`	values of the parameters of the cdf, $F(x)$ specified (as a character string) by `Mixed_dist`.

`tol` the value of ϵ that is used to compute the values of A_i and B_i, $i = 1, ..., n$, as detailed in Step 1 of Section 2.1 in Dimitrova, Kaishev and Tan (2020) (see also (ii) in the Procedure Exact-KS-FFT therein). By default, `tol = 1e-10`. Note that a value of `NA` or `0` will lead to an error!

Details

Given a random sample $\{X_1, ..., X_n\}$ of size `n` with an empirical cdf $F_n(x)$, the Kolmogorov-Smirnov goodness-of-fit statistic is defined as $D_n = \sup |F_n(x) - F(x)|$, where $F(x)$ is the cdf of a prespecified theoretical distribution under the null hypothesis H_0, that $\{X_1, ..., X_n\}$ comes from $F(x)$.

The function `mixed_ks_c_cdf` implements the Exact-KS-FFT method, proposed by Dimitrova, Kaishev, Tan (2020) to compute the complementary cdf $P(D_n \geq q)$ at a value q, when $F(x)$ is mixed. This algorithm ensures a total worst-case run-time of order $O(n^2 log(n))$.

We have not been able to identify alternative, fast and accurate, method (software) that has been developed/implemented when the hypothesized $F(x)$ is mixed.

Value

Numeric value corresponding to $P(D_n \geq q)$.

References

Dimitrina S. Dimitrova, Vladimir K. Kaishev, Senren Tan. (2020) "Computing the Kolmogorov-Smirnov Distribution When the Underlying CDF is Purely Discrete, Mixed or Continuous". Journal of Statistical Software, **95**(10): 1-42. doi:10.18637/jss.v095.i10.

Examples

```
# Compute the complementary cdf of D_{n}
# when the underlying distribution is a mixed distribution
# with two jumps at 0 and log(2.5),
# as in Example 3.1 of Dimitrova, Kaishev, Tan (2020)

## Defining the mixed distribution

Mixed_cdf_example <- function(x)
{
     result <- 0
     if (x < 0){
         result <- 0
     }
     else if (x == 0){
         result <- 0.5
     }
     else if (x < log(2.5)){
         result <- 1 - 0.5 * exp(-x)
     }
     else{
         result <- 1
     }
```

```
     return (result)
 }

KSgeneral::mixed_ks_c_cdf(0.1, 25, c(0, log(2.5)), Mixed_cdf_example)


## Not run:
## Compute P(D_{n} >= q) for n = 5,
## q = 1/5000, 2/5000, ..., 5000/5000
## when the underlying distribution is a mixed distribution
## with four jumps at 0, 0.2, 0.8, 1.0,
## as in Example 2.8 of Dimitrova, Kaishev, Tan (2020)

n <- 5
q <- 1:5000/5000

Mixed_cdf_example <- function(x)
{
  result <- 0
  if (x < 0){
    result <- 0
  }
  else if (x == 0){
    result <- 0.2
  }
  else if (x < 0.2){
    result <- 0.2 + x
  }
  else if (x < 0.8){
    result <- 0.5
  }
  else if (x < 1){
    result <- x - 0.1
  }
  else{
    result <- 1
  }

  return (result)
}

plot(q, sapply(q, function(x) KSgeneral::mixed_ks_c_cdf(x, n,
     c(0, 0.2, 0.8, 1.0), Mixed_cdf_example)), type='l')


## End(Not run)
```

`mixed_ks_test` *Computes the p-value for a one-sample two-sided Kolmogorov-Smirnov test when the cdf under the null hypothesis is mixed*

Description

Computes the p-value $P(D_n \geq d_n)$, where d_n is the value of the KS test statistic computed based on a data sample $\{x_1, ..., x_n\}$, when $F(x)$ is mixed, using the Exact-KS-FFT method expressing the p-value as a double-boundary non-crossing probability for a homogeneous Poisson process, which is then efficiently computed using FFT (see Dimitrova, Kaishev, Tan (2020)).

Usage

```
mixed_ks_test(x, jump_points, Mixed_dist, ..., tol = 1e-10)
```

Arguments

`x`	a numeric vector of data sample values $\{x_1, ..., x_n\}$.
`jump_points`	a numeric vector containing the points of (jump) discontinuity, i.e. where the underlying cdf $F(x)$ has jump(s)
`Mixed_dist`	a pre-specified (user-defined) mixed cdf, $F(x)$, under the null hypothesis.
`...`	values of the parameters of the cdf, $F(x)$ specified (as a character string) by `Mixed_dist`.
`tol`	the value of ϵ that is used to compute the values of A_i and B_i, $i = 1, ..., n$, as detailed in Step 1 of Section 2.1 in Dimitrova, Kaishev and Tan (2020) (see also (ii) in the Procedure Exact-KS-FFT therein). By default, `tol = 1e-10`. Note that a value of `NA` or `0` will lead to an error!

Details

Given a random sample $\{X_1, ..., X_n\}$ of size `n` with an empirical cdf $F_n(x)$, the Kolmogorov-Smirnov goodness-of-fit statistic is defined as $D_n = \sup |F_n(x) - F(x)|$, where $F(x)$ is the cdf of a prespecified theoretical distribution under the null hypothesis H_0, that $\{X_1, ..., X_n\}$ comes from $F(x)$.

The function `mixed_ks_test` implements the Exact-KS-FFT method expressing the p-value as a double-boundary non-crossing probability for a homogeneous Poisson process, which is then efficiently computed using FFT (see Dimitrova, Kaishev, Tan (2020)). This algorithm ensures a total worst-case run-time of order $O(n^2 log(n))$.

The function `mixed_ks_test` computes the p-value $P(D_n \geq d_n)$, where d_n is the value of the KS test statistic computed based on a user-provided data sample $\{x_1, ..., x_n\}$, when $F(x)$ is mixed,

We have not been able to identify alternative, fast and accurate, method (software) that has been developed/implemented when the hypothesized $F(x)$ is mixed.

Value

A list with class "htest" containing the following components:

statistic	the value of the statistic.
p.value	the p-value of the test.
alternative	"two-sided".
data.name	a character string giving the name of the data.

References

Dimitrina S. Dimitrova, Vladimir K. Kaishev, Senren Tan. (2020) "Computing the Kolmogorov-Smirnov Distribution When the Underlying CDF is Purely Discrete, Mixed or Continuous". Journal of Statistical Software, **95**(10): 1-42. doi:10.18637/jss.v095.i10.

Examples

```
# Example to compute the p-value of the one-sample two-sided KS test,
# when the underlying distribution is a mixed distribution
# with two jumps at 0 and log(2.5),
# as in Example 3.1 of Dimitrova, Kaishev, Tan (2020)

# Defining the mixed distribution

Mixed_cdf_example <- function(x)
{
     result <- 0
     if (x < 0){
         result <- 0
     }
     else if (x == 0){
         result <- 0.5
     }
     else if (x < log(2.5)){
         result <- 1 - 0.5 * exp(-x)
     }
     else{
         result <- 1
     }

     return (result)
}
test_data <- c(0,0,0,0,0,0,0.1,0.2,0.3,0.4,
            0.5,0.6,0.7,0.8,log(2.5),log(2.5),
            log(2.5),log(2.5),log(2.5),log(2.5))
KSgeneral::mixed_ks_test(test_data, c(0, log(2.5)),
                         Mixed_cdf_example)


## Compute the p-value of a two-sided K-S test
## when F(x) follows a zero-and-one-inflated
## beta distribution, as in Example 3.3
```

```
## of Dimitrova, Kaishev, Tan (2020)

## The data set is the proportion of inhabitants
## living within a 200 kilometer wide costal strip
## in 232 countries in the year 2010

data("Population_Data")
mu <- 0.6189
phi <- 0.6615
a <- mu * phi
b <- (1 - mu) * phi

Mixed_cdf_example <- function(x)
{
     result <- 0
     if (x < 0){
         result <- 0
     }
     else if (x == 0){
         result <- 0.1141
     }
     else if (x < 1){
         result <- 0.1141 + 0.4795 * pbeta(x, a, b)
     }
     else{
         result <- 1
     }

     return (result)
}
KSgeneral::mixed_ks_test(Population_Data, c(0, 1), Mixed_cdf_example)
```

`Population_Data` *The proportion of inhabitants living within a 200 kilometer wide costal strip in 232 countries in the year 2010*

Description

This data set contains the proportion of inhabitants living within a 200 kilometer wide costal strip in 232 countries in the year 2010. In Example 3.3 of Dimitrova, Kaishev, Tan (2020), the data set is modelled using a zero-and-one-inflated beta distribution in the null hypothesis and a one-sample two-sided Kolmogorov-Smirnov test is performed to test whether the proposed distribution fits the data well enough.

Usage

```
data("Population_Data")
```

Format

A data frame with 232 observations on the proportion of inhabitants living within a 200 kilometer wide costal strip in 2010.

Source

https://sedac.ciesin.columbia.edu/data/set/nagdc-population-landscape-climate-estimates-v3

References

Dimitrina S. Dimitrova, Vladimir K. Kaishev, Senren Tan. (2020) "Computing the Kolmogorov-Smirnov Distribution When the Underlying CDF is Purely Discrete, Mixed or Continuous". Journal of Statistical Software, **95**(10): 1-42. doi:10.18637/jss.v095.i10.

Index

∗ **datasets**
    Population_Data, 38

cont_ks_c_cdf, *4*, *7*, 7, *8*, *25*
cont_ks_cdf, 5, *6*
cont_ks_test, *4*, *9*, 9

disc_ks_c_cdf, *4*, *10*, 10, *11*, *12*, *25*
disc_ks_test, *4*, 13, *14*

ecdf, *11*, *13*

ks.test, *4*, *6*, *7*, *9*, *11*, *12*, *14*, *15*
KS2sample, *4*, 16, *18*, *21*, *23*
KS2sample_c_Rcpp, 19, *21*, *23*
KS2sample_Rcpp, *20*, *21*, 22, *23*
ks_c_cdf_Rcpp, *3*, *4*, *25*, 25, *26*
KSgeneral-package, 2
Kuiper2sample, *4*, *5*, 27, *28*, *30*, *32*
Kuiper2sample_c_Rcpp, 29, *30*, *32*
Kuiper2sample_Rcpp, *30*, 31, *32*

mixed_ks_c_cdf, *4*, *25*, 33, *34*
mixed_ks_test, *4*, 35, *36*

pexp, *9*
pnorm, *9*
Population_Data, 38

stepfun, *11*, *13*