Package 'tnl.Test'

December 7, 2023

Type Package

Title Non-Parametric Tests for the Two-Sample Problem

Version 0.1.0

Description R functions to perform the hypothesis tests for the two sample problem based on order statistics and power comparisons. This package contains functions to calculate the test statistic, density, distribution function, quantile function, random number generation and others. density, distribution function, quantile function, random number generation and others.

License GPL-3

URL https://github.com/ihababusaif/tnl.Test

BugReports https://github.com/ihababusaif/tnl.Test/issues

Imports partitions,
plyr

Suggests covr,
knitr,
rmarkdown,
roxygen2,
testthat (>= 3.0.0)

VignetteBuilder knitr

RdMacros

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

R topics documented:

tnl.test *Non-parametric tests for the two-sample problem based on order statistics and power comparisons*

Description

tnl.test performs a nonparametric test for two sample test on vectors of data.

ptnl gives the distribution function of $T_n^{(\ell)}$ against the specified quantiles.

dtnl gives the density of $T_n^{(\ell)}$ against the specified quantiles.

qtnl gives the quantile function of $T_n^{(\ell)}$ against the specified probabilities.

rtnl generates random values from $T_n^{(\ell)}$.

tnl_mean() gives an expression for $E(T_n^{(\ell)})$under $H_0 : F = G$.

ptnl.lehmann gives the distribution function of $T_n^{(\ell)}$ under Lehmann alternatives.

dtnl.lehmann gives the density of $T_n^{(\ell)}$ under Lehmann alternatives.

qtnl.lehmann gives the quantile function of $T_n^{(\ell)}$ against the specified probabilities under Lehmann alternatives.

rtnl.lehmann generates random values from $T_n^{(\ell)}$ under Lehmann alternatives.

Usage

```
tnl.test(x, y, l, exact = "NULL")

ptnl(q, n, m, l, exact = "NULL", trial = 1e+05)

dtnl(k, n, m, l, exact = "NULL", trial = 1e+05)

qtnl(p, n, m, l, exact = "NULL", trial = 1e+05)

rtnl(N, n, m, l)

tnl_mean(n., m., l)

ptnl.lehmann(q, n., m., l, gamma)

dtnl.lehmann(k, n., m., l, gamma)

qtnl.lehmann(p, n., m., l, gamma)

rtnl.lehmann(N, n., m., l, gamma)
```

Arguments

x	the first (non-empty) numeric vector of data values.
y	the second (non-empty) numeric vector of data values.
l	class parameter of $T_n^{(\ell)}$.

exact	the method that will be used. "NULL" or a logical indicating whether an exact should be computed. See 'Details' for the meaning of NULL.
n, m	samples size.
trial	number of trials for simulation.
k, q	vector of quantiles.
p	vector of probabilities.
N	number of observations. If length(N) > 1, the length is taken to be the number required.
n., m.	samples size.
gamma	parameter of Lehmann alternative.

Details

A non-parametric two-sample test is performed for testing null hypothesis $H_0 : F = G$ against the alternative hypothesis $H_1 : F \neq G$. The assumptions of the $T_n^{(\ell)}$ test are that both samples should come from a continuous distribution and the samples should have the same sample size.
Missing values are silently omitted from x and y.
Exact and simulated p-values are available for the $T_n^{(\ell)}$ test. If exact ="NULL" (the default) the p-value is computed based on exact distribution when the sample size is less than 11. Otherwise, p-value is computed based on a Monte Carlo simulation. If exact ="TRUE", an exact p-value is computed. If exact="FALSE" , a Monte Carlo simulation is performed to compute the p-value. It is recommended to calculate the p-value by a Monte Carlo simulation (use exact="FALSE"), as it takes too long to calculate the exact p-value when the sample size is greater than 10.
The probability mass function (pmf), cumulative density function (cdf) and quantile function of $T_n^{(\ell)}$ are also available in this package, and the above-mentioned conditions about exact ="NULL", exact ="TRUE" and exact="FALSE" is also valid for these functions.
Exact distribution of $T_n^{(\ell)}$ test is also computed under Lehman alternative.
Random number generator of $T_n^{(\ell)}$ test statistic are provided under null hypothesis in the library.

Value

`tnl.test` returns a list with the following components

`statistic:` the value of the test statistic.

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

`ptnl` returns a list with the following components

`method:` The method that was used (exact or simulation). See 'Details'.

`cdf:` distribution function of $T_n^{(\ell)}$ against the specified quantiles.

`dtnl` returns a list with the following components

`method:` The method that was used (exact or simulation). See 'Details'.

`pmf:` density of $T_n^{(\ell)}$ against the specified quantiles.

`qtnl` returns a list with the following components

`method:` The method that was used (exact or simulation). See 'Details'.

`quantile:` quantile function against the specified probabilities.

`rtnl` return *N* of the generated random values.

`tnl_mean()` return the mean of $T_n^{(\ell)}$.

`ptnl.lehmann` return vector of the distribution under Lehmann alternatives against the specified gamma.

`dtnl.lehmann` return vector of the density under Lehmann alternatives against the specified gamma.

`qtnl.lehmann` returns a quantile function against the specified probabilities under Lehmann alternatives.

`rtnl.lehmann` return *N* of the generated random values under Lehmann alternatives.

References

Karakaya K. et al. (2021). *A Class of Non-parametric Tests for the Two-Sample Problem based on Order Statistics and Power Comparisons* . Submitted paper.
Aliev F. et al. (2021). *A Nonparametric Test for the Two-Sample Problem based on Order Statistics*. Submitted paper.

Examples

```
#require(stats)
#x <- rnorm(7, 2, 0.5)
#y <- rnorm(5, 0, 1)
#tnl.test(x, y, l = 2)
## $statistic
## [1] 2
##
## $p.value
## [1]  0.01515152
#ptnl(q = c(2, 5), n = 6, m = 5, l = 2, trial = 100000)
## $method
## [1] "exact"
##
## $cdf
## [1] 0.02164502 1.00000000
#dtnl(k = c(1, 3, 6), n = 7, m = 5, l = 2)
## $method
## [1] "exact"
##
## $pmf
## [1] 0.00000000 0.04671717 0.00000000
#qtnl(p = c(.3, .9), n = 4, m = 5, l = 1)
## $method
## [1] "exact"
##
## $quantile
## [1]  3 4
#rtnl(N = 20, n = 7, m = 10, l = 1)
## [1] 7 6 7 4 6 6 6 5 5 6 6 3 2 4 3 7 7 7 7 6
#require(base)
#tnl_mean(n. = 11, m. = 8, l = 1)
## [1] 5.1693
#ptnl.lehmann(q = 3, n. = 5, m. = 7, l = 2, gamma = 1.2)
## [1] 0.07471397
#dtnl.lehmann(k = 3, n. = 6, m. = 5, l = 2, gamma = 0.8)
## [1] 0.07073467
```

```
#qtnl.lehmann(p = c(.1, .5, .9), n. = 7, m. = 5, l = 1, gamma = 0.5)
## [1] 1 3 5
# rtnl.lehmann(N = 15, n = 7,m=7, l = 2, gamma = 0.5)
## [1] 4 4 7 5 3 7 7 4 4 5 5 7 7 3
```

Index