

Package ‘R2sample’

September 6, 2026

Title Various Methods for the Two Sample Problem

Version 5.0.0

Description The routine `twosample_test()` in this package runs the two sample test using various test statistic. The p values are found via permutation or large sample theory. The routine `twosample_power()` allows the calculation of the power in various cases, and `plot_power()` draws the corresponding power graphs. The routine `run.studies` allows a user to quickly study the power of a new method and how it compares to some of the standard ones.

License GPL (>= 2)

Encoding UTF-8

LinkingTo Rcpp

Imports Rcpp, parallel, shiny, ggplot2, stats, graphics,
microbenchmark, stats

Suggests rmarkdown, knitr, testthat (>= 3.0.0)

VignetteBuilder knitr

Depends R (>= 3.5)

LazyData true

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

NeedsCompilation yes

Author Wolfgang Rolke [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-3514-726X>>)

Maintainer Wolfgang Rolke <wolfgang.rolke@upr.edu>

Repository CRAN

Date/Publication 2026-09-06 15:40:02 UTC

Contents

asymptotic_pvalues	2
case_studies	3

chi_power	4
funcs_list	5
myTS2	5
myTS_cont	6
myTS_disc	6
param	7
plot_power	7
powerR	8
power_newtest	9
power_study	9
pvaluecdf	10
run.studies	10
run_shiny	11
signif_digits	12
timecheck	12
twosample_power	13
twosample_test	14
twosample_test_adjusted_pvalue	16
Index	18

asymptotic_pvalues	<i>This function finds the p values of several tests based on large sample theory</i>
--------------------	---

Description

This function finds the p values of several tests based on large sample theory

Usage

```
asymptotic_pvalues(x, n, m)
```

Arguments

- x a vector of test statistics
- n size of sample 1
- m size of sample 2

Value

A vector of p values.

case_studies	<i>Create case studies</i>
--------------	----------------------------

Description

This function creates the functions needed to run the various case studies. That means it generates two data sets that are equal with respect to three out of these four: mean, variance, skewness and kurtosis, and differ in the other. If `which_diff=mixed`, all are different.

Usage

```
case_studies(  
  which,  
  which_diff = "mean",  
  data_type = "cont",  
  n = 500,  
  nbins = 50,  
  funs_list,  
  ReturnCaseNames = FALSE  
)
```

Arguments

<code>which</code>	name of the case study.
<code>which_diff</code>	="mean", name of parameter in which data sets differ.
<code>data_type</code>	="cont", is data continuous or discrete?
<code>n</code>	=500, sample sizes.
<code>nbins</code>	=50, number of bins for discrete data
<code>funs_list</code>	list of alternative functions
<code>ReturnCaseNames</code>	=FALSE, should list of case studies be returned?

Value

a function that generates data sets

`chi_power`*This function runs the chi-square test for continuous or discrete data*

Description

This function runs the chi-square test for continuous or discrete data

Usage

```
chi_power(  
  rxy,  
  alpha = 0.05,  
  B = 1000,  
  xparam,  
  yparam,  
  nbins = c(50, 10),  
  minexpcount = 5,  
  typeTS  
)
```

Arguments

<code>rx</code>	a function to generate data
<code>alpha</code>	=0.05 type I error probability of test
<code>B</code>	=1000 number of simulation runs
<code>xparam</code>	vector of parameter values
<code>yparam</code>	vector of parameter values
<code>nbins</code>	=c(50, 10) number of desired bins
<code>minexpcount</code>	=5 smallest number of counts required in each bin
<code>typeTS</code>	type of problem, continuous/discrete, with/without weights

Value

A matrix of power values

funcs_list	<i>funcs_list</i>
------------	-------------------

Description

function definitions for case studies

Usage

funcs_list

Format

'funcs_list':
A list of functions

myTS2	<i>a local function needed for the vignette</i>
-------	---

Description

a local function needed for the vignette

Usage

myTS2(x, y, vals)

Arguments

- | | |
|------|--|
| x | An integer vector. |
| y | An integer vector. |
| vals | A numeric vector with the values of the discrete rv. |

Value

A vector with test statistics

myTS_cont	<i>New test for continuous data</i>
-----------	-------------------------------------

Description

This function runs a test for continuous data

Usage

```
myTS_cont(x, y, TSextra)
```

Arguments

x	first data set
y	second data set
TSextra	list with extra info

Value

a number

myTS_disc	<i>This function runs a test for discrete data</i>
-----------	--

Description

This function runs a test for discrete data

Usage

```
myTS_disc(x, y, vals, TSextra)
```

Arguments

x	counts for first data set
y	counts for second data set
vals	values of discrete data
TSextra	list with extra info

Value

a number

param	<i>param</i>
-------	--------------

Description

list of matrices with parameter values for case studies

Usage

```
param
```

Format

'param':

A list of matrices with numbers

plot_power	<i>This function draws the power graph, with curves sorted by the mean power and smoothed for easier reading.</i>
------------	---

Description

This function draws the power graph, with curves sorted by the mean power and smoothed for easier reading.

Usage

```
plot_power(pwr, xname = " ", title = " ", Smooth = TRUE, span = 0.25)
```

Arguments

pwr	a matrix of power values, usually from the twosample_power command
xname	Name of variable on x axis
title	(Optional) title of graph
Smooth	=TRUE lines are smoothed for easier reading
span	=0.25bandwidth of smoothing method

Value

plt, an object of class ggplot.

powerR

Find the power of two sample tests using Rcpp and parallel computing.

Description

Find the power of two sample tests using Rcpp and parallel computing.

Usage

```
powerR(
  rxy,
  xparam,
  yparam,
  TS,
  typeTS,
  TSextra,
  alpha = 0.05,
  B = 1000,
  SuppressMessages,
  maxProcessor
)
```

Arguments

rx	function to generate a list with data sets x, y and (optional) vals, weights
xparam	first argument passed to rx
yparam	second argument passed to rx
TS	test statistic
typeTS	which format has TS?
TSextra	list of items passed TS
alpha	=0.05, the level of the hypothesis test
B	= 1000 number of simulation runs
SuppressMessages	= FALSE print informative messages?
maxProcessor	maximum number of cores to use. If maxProcessor=1 no parallel computing is used.

Value

A numeric vector of power values.

power_newtest	<i>Power for tests with p values</i>
---------------	--------------------------------------

Description

This function estimates the power of test routines that calculate p value(s)

Usage

```
power_newtest(TS, f, xparam, yparam, TSextra, alpha = 0.05, B = 1000)
```

Arguments

TS	routine to calculate test statistics.
f	routine that generates data.
xparam	values of first parameter under the alternative hypothesis.
yparam	values of second parameter under the alternative hypothesis.
TSextra	list passed to TS.
alpha	=0.05 type I error.
B	= 1000 number of simulation runs to estimate the power.

Value

A matrix of power values

power_study	<i>power_study</i>
-------------	--------------------

Description

the results of the included power studies

Usage

```
power_study
```

Format

'power_study':

A list of matrices with powers

pvaluecdf	<i>pvaluecdf</i>
-----------	------------------

Description

data to draw a graph in vignette

Usage

pvaluecdf

Format

'pvaluecdf':

A matrix

run.studies	<i>Power Comparisons</i>
-------------	--------------------------

Description

This function runs the case studies included in the package and compares the power of a new test to those included.

Usage

```
run.studies(
  studies = 1:40,
  types = 1:5,
  data_type = "cont",
  TS,
  TSextra,
  With.p.value = FALSE,
  n = 500,
  alpha = 0.05,
  B = 1000,
  SuppressMessages = TRUE
)
```

Arguments

studies	=1:40, either the names of the studies, or their numbers. If missing all the studies are run.
types	=1:5, either the types of studies, or their numbers. If missing all types are run.
data_type	="cont", continuous or discrete data
TS	routine to calculate test statistics.
TSextra	list passed to TS.
With.p.value	=FALSE does user supplied routine return p values?
n	= 500, desired sample size.
alpha	=0.05 type I error
B	= 1000
SuppressMessages	= TRUE

Details

For details consult vignette("R2sample", "R2sample")

Value

A matrix of power values.

Examples

```
TSextra=list(nbins=5, statistic=FALSE) # Use 5 bins and calculate p values
run.studies(1:2, 1:2, TS=myTS_cont, TSextra=TSextra, With.p.value=TRUE, B=100)
```

run_shiny

Runs the shiny app associated with R2sample package

Description

Runs the shiny app associated with R2sample package

Usage

```
run_shiny()
```

Value

No return value, called for side effect of opening a shiny app

signif_digits*This function does some rounding to nice numbers*

Description

This function does some rounding to nice numbers

Usage

```
signif_digits(x, d = 4)
```

Arguments

x	a list of two vectors
d	=4 number of digits to round to

Value

A list with rounded vectors

timecheck*test function*

Description

test function

Usage

```
timecheck(dta, TS, typeTS, TSextra)
```

Arguments

dta	data set
TS	test statistics
typeTS	format of TS
TSextra	additional info TS

Value

Mean computation time

twosample_power

*Power estimation for two-sample methods***Description**

Find the power of various two sample tests using Rcpp and parallel computing.

Usage

```
twosample_power(
  f,
  ...,
  TS,
  TSextra,
  With.p.value = FALSE,
  alpha = 0.05,
  B = 1000,
  nbins = c(50, 10),
  minexpcount = 5,
  UseLargeSample,
  samplingmethod = "Binomial",
  rnull,
  SuppressMessages = FALSE,
  maxProcessor
)
```

Arguments

f	function to generate a list with data sets x, y and (optional) vals, weights
...	additional arguments passed to f, up to 2
TS	routine to calculate test statistics for non-chi-square tests
TSextra	additional info passed to TS, if necessary
With.p.value	=FALSE does user supplied routine return p values?
alpha	=0.05, the level of the hypothesis test
B	=1000, number of simulation runs.
nbins	=c(50,10), number of bins for chi large and chi small.
minexpcount	=5 minimum required count for chi square tests
UseLargeSample	should p values be found via large sample theory if n,m>10000?
samplingmethod	"Binomial" or independence in discrete data case
rnull	a function that generates data from a model, possibly with parameter estimation.
SuppressMessages	= FALSE print informative messages?
maxProcessor	maximum number of cores to use. If maxProcessor=1 no parallel computing is used.

Details

For details consult vignette("R2sample","R2sample")

This routine runs a number of different two-sample tests for univariate data, either discrete or continuous. The user can also provide their own test method.

Value

A numeric vector of power values.

Examples

```
# Power of standard normal vs. normal with mean mu.
f1=function(mu) list(x=rnorm(25), y=rnorm(25, mu))
#Power of uniform discrete distribution vs. with different probabilities.
twosample_power(f1, mu=c(0,2), B=100, maxProcessor = 1)
f2=function(n, p) list(x=table(sample(1:5, size=1000, replace=TRUE)),
  y=table(sample(1:5, size=n, replace=TRUE,
    prob=c(1, 1, 1, 1, p))), vals=1:5)
twosample_power(f2, n=c(1000, 2000), p=c(1, 1.5), B=100, maxProcessor = 1)
# Compare power of a new test with those in package:
myTS=function(x,y) {z=c(mean(x)-mean(y),sd(x)-sd(y));names(z)=c("M","S");z}
cbind(twosample_power(f1, mu=c(0,2), TS=myTS,B=100, maxProcessor = 1),
  twosample_power(f1, mu=c(0,2), B=100, maxProcessor = 1))
# Power estimation if routine returns a p value, using two parameters for
# the alternative
f3=function(mu, n) list(x=rnorm(25), y=rnorm(n, mu))
myTS2=function(x, y) {out=ks.test(x,y)$p.value; names(out)="KSp"; out}
twosample_power(f3, c(0,1), c(25, 30), TS=myTS2, With.p.value = TRUE, B=100)
```

twosample_test

Tests for the univariate two-sample problem

Description

This function runs a number of two sample tests using Rcpp and parallel computing.

Usage

```
twosample_test(
  x,
  y,
  vals = NA,
  TS,
  TSextra,
  wx = rep(1, length(x)),
  wy = rep(1, length(y)),
  B = 5000,
```

```

    nbins = c(50, 10),
    minexpcount = 5,
    maxProcessor,
    UseLargeSample,
    samplingmethod = "Binomial",
    rnull,
    SuppressMessages = FALSE,
    doMethods = "all"
  )

```

Arguments

x	a vector of numbers if data is continuous or of counts if data is discrete or a list with the data
y	a vector of numbers if data is continuous or of counts if data is discrete.
vals	=NA, a vector of numbers, the values of a discrete random variable. NA if data is continuous data.
TS	routine to calculate test statistics for non-chi-square tests
TSextra	additional info passed to TS, if necessary
wx	A numeric vector of weights of x.
wy	A numeric vector of weights of y.
B	=5000, number of simulation runs for permutation test
nbins	=c(50,10), number of bins for chi square tests.
minexpcount	=5, minimum required expected counts for chi-square tests.
maxProcessor	maximum number of cores to use. If missing (the default) no parallel processing is used.
UseLargeSample	should p values be found via large sample theory if n,m>10000?
samplingmethod	="Binomial" or "independence" for discrete data
rnull	a function that generates data from a model, possibly with parameter estimation.
SuppressMessages	= FALSE print informative messages?
doMethods	="all" a vector of codes for the methods to include. If "all", all methods are used.

Details

For details consult `vignette("R2sample","R2sample")`

Value

A list of two numeric vectors, the test statistics and the p values.

Examples

```
R2sample::twosample_test(rnorm(1000), rt(1000, 4), B=1000)
myTS=function(x,y) {z=c(mean(x)-mean(y),sd(x)-sd(y));names(z)=c("M","S");z}
R2sample::twosample_test(rnorm(1000), rt(1000, 4), TS=myTS, B=1000)
vals=1:5
x=table(sample(vals, size=100, replace=TRUE))
y=table(sample(vals, size=100, replace=TRUE, prob=c(1,1,3,1,1)))
R2sample::twosample_test(x, y, vals)
```

twosample_test_adjusted_pvalue

Adjusted p values for simultaneous testing in the two-sample problem.

Description

This function runs a number of two sample tests using Rcpp and parallel computing and then finds the correct p value for the combined tests.

Usage

```
twosample_test_adjusted_pvalue(
  x,
  y,
  vals = NA,
  TS,
  TSextra,
  wx = rep(1, length(x)),
  wy = rep(1, length(y)),
  B = c(5000, 1000),
  nbins = c(50, 10),
  minexpcount = 5,
  samplingmethod = "independence",
  rnull,
  SuppressMessages = FALSE,
  doMethods
)
```

Arguments

x	a vector of numbers if data is continuous or of counts if data is discrete, or a list with the data.
y	a vector of numbers if data is continuous or of counts if data is discrete.
vals	=NA, a vector of numbers, the values of a discrete random variable. NA if data is continuous data.
TS	routine to calculate test statistics for non-chi-square tests
TSextra	additional info passed to TS, if necessary

wx	A numeric vector of weights of x.
wy	A numeric vector of weights of y.
B	=c(5000, 1000), number of simulation runs for permutation test
nbins	=c(50,10), number of bins for chi square tests.
minexpcount	= 5, minimum required expected counts for chi-square tests
samplingmethod	"independence" or "Binomial" for discrete data
rnull	routine for parametric bootstrap
SuppressMessages	= FALSE print informative messages?
doMethods	"all" a vector of codes for the methods to include. If "all", all methods are used.

Details

For details consult vignette("R2sample","R2sample")

Value

A list of two numeric vectors, the test statistics and the p values.

Examples

```
x=rnorm(100)
y=rt(200, 4)
R2sample::twosample_test_adjusted_pvalue(x, y, B=c(500, 500))
vals=1:5
x=table(c(1:5, sample(1:5, size=100, replace=TRUE)))-1
y=table(c(1:5, sample(1:5, size=100, replace=TRUE, prob=c(1,1,3,1,1))))-1
R2sample::twosample_test_adjusted_pvalue(x, y, vals, B=c(500, 500))
```

Index

* datasets

- fun_list, [5](#)
- param, [7](#)
- power_study, [9](#)
- pvaluecdf, [10](#)

asymptotic_pvalues, [2](#)

case_studies, [3](#)

chi_power, [4](#)

fun_list, [5](#)

myTS2, [5](#)

myTS_cont, [6](#)

myTS_disc, [6](#)

param, [7](#)

plot_power, [7](#)

power_newtest, [9](#)

power_study, [9](#)

powerR, [8](#)

pvaluecdf, [10](#)

run.studies, [10](#)

run_shiny, [11](#)

signif_digits, [12](#)

timecheck, [12](#)

twosample_power, [13](#)

twosample_test, [14](#)

twosample_test_adjusted_pvalue, [16](#)