

# Package ‘R2sample’

October 21, 2024

**Title** Various Methods for the Two Sample Problem

**Version** 2.2.0

**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.

**License** GPL (>= 2)

**Encoding** UTF-8

**RoxygenNote** 7.2.1

**LinkingTo** Rcpp

**Imports** Rcpp, parallel, shiny, ggplot2, stats

**Suggests** rmarkdown, knitr

**VignetteBuilder** knitr

**NeedsCompilation** yes

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

**Maintainer** Wolfgang Rolke <[wolfgang.rolke@upr.edu](mailto:wolfgang.rolke@upr.edu)>

**Depends** R (>= 3.5.0)

**Repository** CRAN

**Date/Publication** 2024-10-21 19:20:02 UTC

## Contents

<code>asymptotic_pvalues</code> . . . . .	2
<code>chi_power</code> . . . . .	2
<code>plot_power</code> . . . . .	3
<code>run_shiny</code> . . . . .	4
<code>signif.digits</code> . . . . .	4
<code>twosample_power</code> . . . . .	5
<code>twosample_test</code> . . . . .	6
<code>twosample_test_adjusted_pvalue</code> . . . . .	7

**Index****9**

---

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.

---

chi_power	<i>This function runs the chi-square test for continuous or discrete data</i>
-----------	---

---

**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**

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

**Value**

A matrix of power values

---

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.

---

run_shiny	<i>Runs the shiny app associated with R2sample package</i>
-----------	--

---

**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	<i>This function does some rounding to nice numbers</i>
---------------	---

---

**Description**

This function does some rounding to nice numbers

**Usage**

```
## S3 method for class 'digits'  
signif(x, d = 4)
```

**Arguments**

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

**Value**

A list with rounded vectors

---

twosample_power	<i>Find the power of various two sample tests using Rcpp and parallel computing.</i>
-----------------	--

---

**Description**

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

**Usage**

```
twosample_power(
  f,
  ...,
  TS,
  TSextra,
  alpha = 0.05,
  B = c(1000, 1000),
  nbins = c(50, 10),
  minexpcount = 5,
  UseLargeSample,
  samplingmethod = "independence",
  maxProcessor = 10
)
```

**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
alpha	=0.05, the level of the hypothesis test
B	=c(1000, 2000), number of simulation runs for power and permutation test.
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	=independence or MCMC in discrete data case
maxProcessor	=10, maximum number of cores to use. If maxProcessor=1 no parallel computing is used.

**Value**

A numeric vector of power values.

**Examples**

```
f=function(mu) list(x=rnorm(25), y=rnorm(25, mu))
twosample_power(f, mu=c(0,2), B=c(100, 100), maxProcessor = 1)
f=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(f, n=c(1000, 2000), p=c(1, 1.5), B=c(100, 100), maxProcessor = 1)
```

---

twosample_test	<i>This function runs a number of two sample tests using Rcpp and parallel computing.</i>
----------------	---

---

**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),
  maxProcessor,
  UseLargeSample,
  samplingmethod = "independence",
  doMethods = "all"
)
```

**Arguments**

x	a vector of numbers if data is continuous or of counts if data is discrete.
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.  
**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** ="independence" or "MCMC" for discrete data  
**doMethods** ="all" Which methods should be included? If missing all methods are used.

### 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

*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.*

---

### 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),
  samplingmethod = "independence",
  doMethods
)
  
```

**Arguments**

<code>x</code>	a vector of numbers if data is continuous or of counts if data is discrete.
<code>y</code>	a vector of numbers if data is continuous or of counts if data is discrete.
<code>vals</code>	=NA, a vector of numbers, the values of a discrete random variable. NA if data is continuous data.
<code>TS</code>	routine to calculate test statistics for non-chi-square tests
<code>TSextra</code>	additional info passed to TS, if necessary
<code>wx</code>	A numeric vector of weights of x.
<code>wy</code>	A numeric vector of weights of y.
<code>B</code>	=c(5000, 1000), number of simulation runs for permutation test
<code>nbins</code>	=c(50,10), number of bins for chi square tests.
<code>samplingmethod</code>	="independence" or "MCMC" for discrete data
<code>doMethods</code>	Which methods should be included?

**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

`asymptotic_pvalues`, [2](#)

`chi_power`, [2](#)

`plot_power`, [3](#)

`run_shiny`, [4](#)

`signif.digits`, [4](#)

`twosample_power`, [5](#)

`twosample_test`, [6](#)

`twosample_test_adjusted_pvalue`, [7](#)