Package ‘R2sample’

October 26, 2022

Title Two Sample Problem Routines using Permutation

Version 1.0.0

Description The routine twosample_test() in this package runs the
two sample test using various test statistic. The p values are
found via permutation. 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, microbenchmark

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>

Repository CRAN

Date/Publication 2022-10-26 10:57:56 UTC

R topics documented:

- bincounter_cpp .................................................. 2
- chi_test_cont_cpp ................................................ 2
- chi_test_disc_cpp ................................................ 3
- permute_cont_cpp ................................................. 3
- permute_disc_cpp ................................................ 4
- perm_test_cpp ..................................................... 4
- plot_power .......................................................... 5
- power_cpp .......................................................... 5
- rep_cpp ............................................................. 6
bincounter_cpp

Count occurrences in bins. Useful for power calculations. Replaces hist command from R.

**Usage**

```
bincounter_cpp(x, bins)
```

**Arguments**

- `x`: numeric vector
- `bins`: numeric vector

**Value**

Integer vector of counts

---

chi_test_cont_cpp

Run chi square test for continuous data.

**Description**

Run chi square test for continuous data.

**Usage**

```
chi_test_cont_cpp(dta, nbins = as.integer(c(100, 10)))
```

**Arguments**

- `dta`: A list of numeric vectors.
- `nbins`: A vector of length 2 of bin lengths.

**Value**

A list with test statistics and p values
**Description**

run chi square test for discrete data.

**Usage**

`chi_test_disc_cpp(dta, nbins = as.integer(c(100, 10)))`

**Arguments**

- `dta` A list of numeric vectors.
- `nbins` Integer vector of length 2 with number of bins

**Value**

A list with test statistics, p values and degrees of freedom

---

**Description**

permute continuous data

**Usage**

`permute_cont_cpp(dta)`

**Arguments**

- `dta` A list of numeric vectors.

**Value**

A list of permuted x and y vectors
permute_disc_cpp  permute discrete data

Description
permute discrete data

Usage
permute_disc_cpp(dta)

Arguments
dta A list of numeric vectors.

Value
A list of permuted vectors

perm_test_cpp  run permutation test.

Description
run permutation test.

Usage
perm_test_cpp(
x, 
y, 
vals = 0, 
nbins = as.integer(c(100, 10)), 
B = 5000L, 
doMethod = as.character(c("chi large", "chi small", "t test", "KS", "Kuiper", "CvM", 
"AD", "LR", "ZA", "ZK", "ZC", "Wassp1"))
)

Arguments
x A numeric vector.
y A numeric vector.
vals A numeric vector. Indicates discrete data.
nbins Two bin numbers for chi square test.
B Number of simulation runs.
doMethod A character vector of methods to include
plot_power

Value
A list with test statistics and p values

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

Usage
plot_power(pwr, xname = "", Smooth = TRUE)

Arguments
pwr a matrix of power values, usually from the twosample_power command
xname Name of variable on x axis
Smooth =TRUE lines are smoothed for easier reading

Value
plt, an object of class ggplot.

power_cpp

Find the power of various tests via permutation.

Description
Find the power of various tests via permutation.

Usage
power_cpp(
  rxy,
  nbins = as.integer(c(100, 10)),
  alpha = 0.05,
  B = 1000L,
  xparam = 0,
  yparam = 0,
  doMethod = as.character(c("chi large", "chi small", "t test", "KS", "Kuiper", "CvM",
                           "AD", "LR", "ZA", "ZK", "ZC", "Wasspl"))
)


Arguments

- `rxy` a function that generates x and y data.
- `nbins` Two bin numbers for chi square test.
- `alpha` A numeric constant
- `B` Number of simulation runs.
- `xparam` arguments for r1.
- `yparam` arguments for r2.
- `doMethod` A character vector of methods to include

Value

A numeric matrix of powers

---

**rep_cpp**

cpp version of R routine rep

---

Description

cpp version of R routine rep

Usage

`rep_cpp(x, times)`

Arguments

- `x` numeric vector
- `times` integer vector

Value

A numeric vector

---

**run_shiny**

run_shiny

---

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
TS_cont_cpp  
*find test statistics for continuous data*

**Description**
find test statistics for continuous data

**Usage**

```r
TS_cont_cpp(
  dta,
  doMethod = as.character(c("chi large", "chi small", "t test", "KS", "Kuiper", "CvM",
                          "AD", "LR", "ZA", "ZK", "ZC", "Wassp1"))
)
```

**Arguments**
- **dta**  
  A list
- **doMethod**  
  A character vector of methods to include

**Value**
A vector of test statistics

---

TS_disc_cpp  
*find test statistics for discrete data*

**Description**
find test statistics for discrete data

**Usage**

```r
TS_disc_cpp(
  dta,
  ADweights,
  doMethod = as.character(c("chi large", "chi small", "t test", "KS", "Kuiper", "CvM",
                          "AD", "LR", "ZA", "ZK", "ZC", "Wassp1"))
)
```

**Arguments**
- **dta**  
  A list
- **ADweights**  
  A vector of weights for AD method
- **doMethod**  
  A character vector of methods to include
twosample_power

Value
A vector of test statistics

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

Usage
twosample_power(
  f,
  ...,
  alpha = 0.05,
  B = 1000,
  nbins = c(100, 10),
  maxProcessor = 10,
  doMethod = "all"
)

Arguments
  f          function to generate a list with data sets x, y and (optional) vals
  ...       additional arguments passed to f
  alpha = 0.05, the level of the hypothesis test
  B = 1000, number of simulation runs for permutation test and power.
  nbins = c(100,10), number of bins for chi large and chi small.
  maxProcessor = 10, maximum number of cores to use. If maxProcessor=1 no parallel computing is used.
  doMethod = "all", which methods should be included?

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=100, maxProcessor = 1)
f=function() list(x=table(sample(1:10, size=1000, replace=TRUE)),
  y=table(sample(1:10, size=1200, replace=TRUE)), vals=1:10)
twosample_power(f, B=100, maxProcessor = 1)
Description

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

Usage

twosample_test(
  x,
  y,
  vals,
  B = 5000,
  nbins = c(100, 10),
  maxProcessor = 10,
  discretize = FALSE,
  doMethod
)

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 a vector of numbers, the values of a discrete random variable. If it is missing, continuous data is assumed.

B =5000, number of simulation runs for permutation test

nbins =c(100,10), number of bins for chi square tests.

maxProcessor =10, maximum number of cores to use. If maxProcessor=1 no parallel computing is used.

discretize =FALSE. Should continuous data be binned?

doMethod Which methods should be included? If missing default methods are used.

Value

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

Examples

twosample_test(rnorm(1000), rt(1000, 4), B=1000, maxProcessor = 1)
vals=1:5
x=table(sample(vals, size=100, replace=TRUE))
y=table(sample(vals, size=100, replace=TRUE, prob=c(1,1,2,1,1)))
twosample_test(x, y, vals, maxProcessor = 1)
weights_cpp

*Description*

find weights for several statistics for discrete data

*Usage*

weights_cpp(dta)

*Arguments*

dta

A list with vectors x, y and vals

*Value*

A vector of weights
Index

bincounter_cpp, 2
chi_test_cont_cpp, 2
chi_test_disc_cpp, 3
perm_test_cpp, 4
permute_cont_cpp, 3
permute_disc_cpp, 4
plot_power, 5
power_cpp, 5
rep_cpp, 6
run_shiny, 6
TS_cont_cpp, 7
TS_disc_cpp, 7
twosample_power, 8
twosample_test, 9
weights_cpp, 10