Package ‘R2sample’

January 23, 2023

Title  Two Sample Problem Routines using Permutation
Version  1.1.0
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  2023-01-23 15:20:02 UTC

R topics documented:

plot_power ................................................. 2
run_shiny .................................................. 2
twosample_power ......................................... 3
twosample_test .......................................... 4

Index 5
plot_power

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

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.

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
twosample_power

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

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
...
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)
twosample_test  

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

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)
Index

plot_power, 2
run_shiny, 2

twosample_power, 3
twosample_test, 4