Package 'MD2sample'

March 28, 2026

Title Various Methods for the Two Sample Problem in D>1 Dimensions

Version 1.2.0

Description The routine twosample_test() in this package runs the
two-sample test using various test statistic for multivariate data.
The user can also run several tests and then find a p value adjusted for simultaneous inference.
The p values are found via permutation or via the parametric bootstrap.
The routine twosample_power() allows the estimation of the power of the tests.
The routine run.studies() allows a user to quickly study the power of a new method and
how it compares to those included in the package.
For details of the methods and references see the included vignettes.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.3.3

LinkingTo Rcpp

Imports Rcpp, parallel, stats, FNN, copula, ade4, gTests, igraph, lsa,
microbenchmark, mvtnorm, Ball

Suggests rmarkdown, knitr, ggplot2, egg, testthat (>= 3.0.0)

VignetteBuilder knitr

Depends R (>= 3.5)

LazyData true

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 2026-03-28 06:50:06 UTC

Contents

`case.studies` *Create case studies*

Description

This function creates the functions needed to run the various case studies.

Usage

```
case.studies(
  which,
  n = 200,
  nx = n,
  ny = n,
  nbins = -1,
  Ranges = matrix(c(-Inf, Inf, -Inf, Inf), 2, 2),
  ReturnCaseNames = FALSE
)
```

Arguments

`which`	name of the case study.
`n`	=200, sample size for both data sets
`nx`	=n, sample size for first data sets
`ny`	=n, sample size for second data sets
`nbins`	=-1, number of bins for chi-square test, 2D only
`Ranges`	= matrix(c(-Inf, Inf, -Inf, Inf), 2, 2), ranges of variables
`ReturnCaseNames`	=FALSE, should list of case studies be returned?

Value

a list of functions and vectors

`chisq2D_test_cont` *This function does the chi square two sample test for continuous data in two dimensions*

Description

This function does the chi square two sample test for continuous data in two dimensions

Usage

```
chisq2D_test_cont(
  dta_x,
  dta_y,
  Ranges = matrix(c(-Inf, Inf, -Inf, Inf), 2, 2),
  nbins = c(5, 5),
  minexpcount = 5,
  SuppressMessages = FALSE
)
```

Arguments

`dta_x`	a matrix of numbers
`dta_y`	a matrix of numbers
`Ranges`	=matrix(c(-Inf, Inf, -Inf, Inf),2,2) a 2x2 matrix with lower and upper bounds
`nbins`	=c(5,5) number of bins in x and y direction
`minexpcount`	=5 minimum counts required per bin
`SuppressMessages`	=FALSE, print informative messages?

Value

a list with statistics and p values

chisq2D_test_disc — *This function does the chi square two sample test for continuous data in two dimensions*

Description

This function does the chi square two sample test for continuous data in two dimensions

Usage

```
chisq2D_test_disc(dta, minexpcount = 5)
```

Arguments

dta	a list with values and counts
minexpcount	=5 minimum counts required per bin

Value

a list with statistics and p values

chiTS.cont — *Chi-square test for continuous data, example of user supplied function*

Description

This function does the chi square two sample test for continuous data in two dimensions

Usage

```
chiTS.cont(x, y, TSextra)
```

Arguments

x	a matrix of numbers
y	a matrix of numbers
TSextra	list with some info

Value

either a test statistic or a p value

chiTS.disc *Chi-square test for discrete data, example of user supplied function*

Description

This function does the chi square two sample test for discrete data in two dimensions

Usage

```
chiTS.disc(x, y, vals_x, vals_y, TSextra)
```

Arguments

x	a vector with counts
y	a vector with counts
vals_x	a vector with points
vals_y	a vector with points
TSextra	list with some info

Value

either a test statistic or a p value

edge.tests *This function does the tests by Chen and Friedman*

Description

This function does the tests by Chen and Friedman

Usage

```
edge.tests(x, y)
```

Arguments

x	a matrix of numbers
y	a matrix of numbers

Value

a list with statistics and p values

FR.test *Friedman-Rafsky (FR) test*

Description

FR test is a multivariate generalization of nonparametric two-sample test. This function is an implementation with customized options, including a visualization of the minimum spanning tree (MST).

Usage

```
FR.test(
  samp1,
  samp2,
  use.cosine = FALSE,
  binary = FALSE,
  binary.cutoff = 2,
  plot.MST = FALSE,
  col = c("#F0E442", "#56B4E9"),
  label.names = c("Sample 1", "Sample 2"),
  vertex.size = 5,
  edge.width = 1,
  ...
)
```

Arguments

samp1	Numeric matrix or data frame for Sample 1. Rows are multivariate dimensions, and columns are samples. E.g. gene by cell.
samp2	Numeric matrix or data frame for Sample 2.
use.cosine	An option if to use cosine distance. Logical variable. By default (FALSE), Euclidean distance is used.
binary	An option if to use binary values. Logical variable. Default: FALSE. If TRUE, use binary.cutoff to dichotomize samp1 and samp2.
binary.cutoff	Numeric value for binary cutoff. Binary value = 1 if greater than binary.cutoff, 0 otherwise. Default: 2.
plot.MST	Boolean variable indicating if to plot the minimum spanning tree (MST). Default: FALSE.
col	Character vector of length two for customized colors of the nodes in MST. Default: c("#F0E442", "#56B4E9").
label.names	Character vector of length two for customized names of the two samples. Default: c("Sample 1","Sample 2").
vertex.size, edge.width, ...	Additional plotting parammeters passed to plot.igraph. Default: vertex.size=5, edge.width=1.

Value

Test statistics and p-values.

`runs`	Total number of subtrees.
`runs.samp1`	Number of subtrees of Sample 1.
`runs.samp2`	Number of subtrees of Sample 2.
`stat`	The standardized FR statistic.
`p.value`	P-value of the FR test.

`power_studies_results` *power_studies_results*

Description

the results of the included power studies

Usage

```
power_studies_results
```

Format

'power_studies_results':
A list of matrices with powers

`run.studies` *Benchmarking for Multivariate Two-Sample Tests*

Description

This function runs the case studies included in the package.

Usage

```
run.studies(
  Continuous = TRUE,
  study,
  TS,
  TSextra,
  With.p.value = FALSE,
  nsample = 200,
  alpha = 0.05,
  param_alt,
  SuppressMessages = FALSE,
  B = 1000,
  maxProcessor
)
```

Arguments

Continuous	=TRUE, run cases for continuous data.
study	either the name of the study, or its number in the list. If missing all the studies are run.
TS	routine to calculate new test statistics.
TSextra	list passed to TS (optional).
With.p.value	=FALSE, does user supplied routine return p values?
nsample	= 200, desired sample size. 200 is used in included case studies.
alpha	=0.05, type I error probability of tests. 0.05 is used in included case studies.
param_alt	(list of) values of parameter under the alternative hypothesis. If missing included values are used.
SuppressMessages	=FALSE, should informative messages be printed?
B	= 1000, number of simulation runs.
maxProcessor	number of cores to use. If missing the number of physical cores-1 is used. If set to 1 no parallel processing is done.

Details

For details consult vignette(package="MD2sample")

Value

A (list of) matrices of p.values.

Examples

```
#The new test is a (included) chi square test:
TSextra=list(which="pval", nbins=rbind(c(3,3), c(4,4)))
run.studies(Continuous=TRUE, study=c("NormalD2", "tD2"),
          TS=MD2sample::chiTS.cont, TSextra=TSextra, B=100)
```

---

signif.digits *This function does some rounding to nice numbers*

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

`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

`TS_cont_pval` *This function runs a number of two sample tests which find their own p value.*

Description

This function runs a number of two sample tests which find their own p value.

Usage

```
TS_cont_pval(x, y)
```

Arguments

x	a matrix of numbers if data is continuous or of counts if data is discrete.
y	a matrix of numbers if data is continuous or of counts if data is discrete.

Value

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

---

twosample_power *Power Estimation for Multivariate Two-Sample Tests*

Description

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

Usage

```
twosample_power(
  f,
  ...,
  TS,
  TSextra,
  alpha = 0.05,
  B = 1000,
  nbins = c(5, 5),
  minexpcount = 5,
  Ranges = matrix(c(-Inf, Inf, -Inf, Inf), 2, 2),
  samplingmethod = "Binomial",
  rnull,
  With.p.value = FALSE,
  DoTransform = TRUE,
  SuppressMessages = FALSE,
  LargeSampleOnly = FALSE,
  maxProcessor,
  doMethods = "all"
)
```

Arguments

f	function to generate a list with data sets x and y for continuous data or a matrix with columns vals_x, vals_y, x and y for discrete data.
...	additional arguments passed to f, up to 2.
TS	routine to calculate test statistics for new tests.
TSextra	additional info passed to TS, if necessary.
alpha	=0.05, the type I error probability of the hypothesis test.

B	=1000, number of simulation runs.
nbins	=c(5, 5), number of bins for chi square test if Dim=2.
minexpcount	=5, lowest required count for chi-square test.
Ranges	=matrix(c(-Inf, Inf, -Inf, Inf),2,2), a 2x2 matrix with lower and upper bounds.
samplingmethod	="Binomial" for Binomial sampling or "independence" for independence sampling in the discrete data case.
rnull	function to generate new data sets for parametric bootstrap.
With.p.value	=FALSE, does user supplied routine return p values?
DoTransform	=TRUE, should data be transformed to to unit hypercube?
SuppressMessages	=FALSE, should messages be printed?
LargeSampleOnly	=FALSE, should only methods with large sample theories be run?
maxProcessor	number of cores to use. If missing the number of physical cores-1 is used. If set to 1 no parallel processing is done.
doMethods	="all", which methods should be included?

Details

For details consult vignette("MD2sample","MD2sample")

Value

A numeric matrix or vector of power values.

Examples

```
#Note that the resulting power estimates are meaningless because
#of the extremely low number of simulation runs B, required because of CRAN timing rule
#
#Power of tests when one data set comes from a standard normal multivariate distribution function
#and the other data set from a multivariate normal with correlation
#number of simulation runs is ridiculously small because of CRAN submission rules
f=function(a=0) {
 S=diag(2)
 x=mvtnorm::rmvnorm(100, sigma = S)
 S[1,2]=a
 S[2,1]=a
 y=mvtnorm::rmvnorm(120, sigma = S)
 list(x=x, y=y)
}
twosample_power(f, c(0, 0.5), B=10, maxProcessor=1)
#Power of use supplied test. Example is a (included) chi-square test:
TSextra=list(which="statistics", nbins=rbind(c(3,3), c(4,4)))
twosample_power(f, c(0, 0.5), TS=chiTS.cont, TSextra=TSextra, B=10, maxProcessor=1)
#Same example, but this time the user supplied routine calculates p values:
TSextra=list(which="pvalues", nbins=c(4,4))
twosample_power(f, c(0, 0.5), TS=chiTS.cont, TSextra=TSextra, B=10,
```

```
            With.p.value=TRUE, maxProcessor=1)
#Example for discrete data
g=function(p1, p2) {
  x = table(sample(1:4, size=1000, replace = TRUE))
  y = table(sample(1:4, size=500, replace = TRUE, prob=c(p1,p2,1,1)))
  cbind(vals_x=rep(1:2,2),  vals_y=rep(1:2, each=2), x=x, y=y)
}
twosample_power(g, 1.5, 1.6, B=10, maxProcessor=1)
```

`twosample_test` *Multivariate Two-Sample Tests*

Description

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

Usage

```
twosample_test(
  x,
  y,
  vals_x = NA,
  vals_y = NA,
  TS,
  TSextra,
  B = 5000,
  nbins = c(5, 5),
  minexpcount = 5,
  Ranges = matrix(c(-Inf, Inf, -Inf, Inf), 2, 2),
  DoTransform = TRUE,
  samplingmethod = "Binomial",
  rnull,
  SuppressMessages = FALSE,
  LargeSampleOnly = FALSE,
  maxProcessor,
  doMethods = "all"
)
```

Arguments

x	Continuous data: either a matrix of numbers, or a list with two matrices called x and y. if it is a matrix Observations are in different rows. Discrete data: a vector of counts or a matrix with columns named vals_x, vals_y, x and y.
y	a matrix of numbers if data is continuous or a vector of counts if data is discrete.
vals_x	=NA, a vector of values for discrete random variables, or NA if data is continuous.

vals_y	=NA, a vector of values for discrete random variables, or NA if data is continuous.
TS	user supplied routine to calculate test statistics for new tests.
TSextra	(optional) additional info passed to TS, if necessary.
B	=5000, number of simulation runs for permutation test.
nbins	=c(5,5), for chi square tests (2D only).
minexpcount	=5, lowest required count for chi-square test (2D only).
Ranges	=matrix(c(-Inf, Inf, -Inf, Inf),2,2), a 2x2 matrix with lower and upper bounds (2D only).
DoTransform	=TRUE, should data be transformed to unit hypercube?
samplingmethod	="Binomial" for Binomial sampling or "independence" for independence sampling.
rnull	function to generate new data sets for simulation as an alternative to the permutation method.
SuppressMessages	=FALSE, should informative messages be printed?
LargeSampleOnly	=FALSE, should only methods with large sample theories be run?
maxProcessor	number of cores to use. If missing the number of physical cores-1 is used. If set to 1 no parallel processing is done.
doMethods	="all", Which methods should be included?

Details

For details consult vignette("MD2sample","MD2sample")

Value

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

Examples

```
#Two continuous data sets from a multivariate normal:
x = mvtnorm::rmvnorm(100, c(0,0))
y = mvtnorm::rmvnorm(120, c(0,0))
twosample_test(x, y, B=100, maxProcessor=1)
#Using a new test, this one is an (included) chi square test.
#Also enter data as a list:
TSextra=list(which="statistics", nbins=rbind(c(3,3), c(4,4)))
dta=list(x=x, y=y)
twosample_test(dta, TS=chiTS.cont, TSextra=TSextra, B=100, maxProcessor=1)
#Two discrete data sets from some distribution:
x = table(sample(1:4, size=1000, replace = TRUE))
y = table(sample(1:4, size=1000, replace = TRUE, prob=c(1,2,1,1)))
vals_x=rep(1:2,2)
vals_y=rep(1:2, each=2)
twosample_test(x, y, vals_x, vals_y, B=100, maxProcessor=1)
```

```
#Run a discrete chi square test and enter the data as a matrix:
TSextra=list(which="statistics")
dta=cbind(x=x, y=y, vals_x=vals_x, vals_y=vals_y)
twosample_test(dta, TS=chiTS.disc, TSextra=TSextra, B=100, maxProcessor=1)
```

`twosample_test_adjusted_pvalue` *Adjusted p values*

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_x = NA,
  vals_y = NA,
  B = c(5000, 1000),
  nbins = c(5, 5),
  minexpcount = 5,
  samplingmethod = "Binomial",
  Ranges = matrix(c(-Inf, Inf, -Inf, Inf), 2, 2),
  DoTransform = TRUE,
  rnull,
  SuppressMessages = FALSE,
  maxProcessor,
  doMethods
)
```

Arguments

`x`	Continuous data: either a matrix of numbers, or a list with two matrices called x and y. if it is a matrix Observations are in different rows. Discrete data: a vector of counts or a matrix with columns named vals_x, vals_y, x and y.
`y`	a matrix of numbers if data if data is continuous or a vector of counts if data is discrete.
`vals_x`	=NA, a vector of values for discrete random variable, or NA if data is continuous.
`vals_y`	=NA, a vector of values for discrete random variable, or NA if data is continuous.
`B`	=c(5000, 1000), number of simulation runs for permutation test and for estimation of the empirical distribution function.
`nbins`	=c(5, 5), number of bins for chi square tests (2D only).
`minexpcount`	= 5, minimum required expected counts for chi-square tests.

samplingmethod	="Binomial" or "independence" for discrete data.
Ranges	=matrix(c(-Inf, Inf, -Inf, Inf),2,2) a 2x2 matrix with lower and upper bounds.
DoTransform	=TRUE, should data be transformed to interval (0,1)?
rnull	routine for parametric bootstrap.
SuppressMessages	= FALSE, print informative messages?
maxProcessor	number of cores for parallel processing.
doMethods	Which methods should be included? If missing a small number of methods that generally have good power are used.

Details

For details consult the vignette("MD2sample","MD2sample")

Value

NULL, results are printed out.

Examples

```
#Note that the number of simulation runs B is very small to
#satisfy CRAN's run time constraints.
#Two continuous data sets from a multivariate normal:
x = mvtnorm::rmvnorm(100, c(0,0))
y = mvtnorm::rmvnorm(120, c(0,0))
twosample_test_adjusted_pvalue(x, y, maxProcessor=1, B=20)
#Two discrete data sets from some distribution:
x = table(sample(1:4, size=1000, replace = TRUE))
y = table(sample(1:4, size=500, replace = TRUE, prob=c(1, 1.5, 1, 1)))
twosample_test_adjusted_pvalue(x, y, rep(1:2,2), rep(1:2, each=2), maxProcessor=1, B=20)
```

Index