Package ‘SHT’

November 3, 2022

Type     Package
Title    Statistical Hypothesis Testing Toolbox
Version  0.1.8
Description We provide a collection of statistical hypothesis testing procedures ranging from classical to modern methods for non-trivial settings such as high-dimensional scenario. For the general treatment of statistical hypothesis testing, see the book by Lehmann and Romano (2005) <doi:10.1007/0-387-27605-X>.
License MIT + file LICENSE
Encoding UTF-8
URL https://www.kisungyou.com/SHT/
BugReports https://github.com/kisungyou/SHT/issues
Imports Rcpp, Rdpack, stats, utils, pracma, flare
LinkingTo Rcpp, RcppArmadillo
RoxygenNote 7.2.1
RdMacros Rdpack
NeedsCompilation yes
Author Kyoungjae Lee [aut],
        Lizhen Lin [aut],
        Kisung You [aut, cre] (<https://orcid.org/0000-0002-8584-459X>)
Maintainer Kisung You <kisungyou@outlook.com>
Repository CRAN
Date/Publication 2022-11-03 08:26:47 UTC

R topics documented:

cov1.2012Fisher .................................................. 3
 cov1.2015WL .................................................. 4
 cov2.2012LC .................................................. 5
 cov2.2013CLX ................................................ 7
 cov2.2015WL ................................................ 8
R topics documented:

covk.2001Schott ................................................... 9  
covk.2007Schott .................................................. 11  
eqdist.2014BG ..................................................... 12  
mean1.1931Hotelling ............................................ 14  
mean1.1958Dempster ............................................. 15  
mean1.1996BS ................................................... 16  
mean1.2008SD ................................................... 17  
mean1.ttest ...................................................... 18  
mean2.1931Hotelling ............................................ 20  
mean2.1958Dempster ............................................. 21  
mean2.1965Yao .................................................. 22  
mean2.1980Johansen ............................................. 24  
mean2.1986NVM ................................................ 25  
mean2.1996BS .................................................. 26  
mean2.2004KY .................................................. 28  
mean2.2008SD .................................................. 29  
mean2.2011LJW ................................................ 30  
mean2.2014CLX ................................................ 32  
mean2.2014Thulin ............................................... 33  
mean2.mxPBF .................................................. 35  
mean2.ttest ...................................................... 36  
meank.2007Schott .............................................. 37  
meank.2009ZX .................................................. 39  
meank.2019CPH ................................................ 40  
meank.anova ................................................... 42  
mvar1.1998AS .................................................. 43  
mvar1.LRT ..................................................... 44  
mvar2.1930PN ................................................ 45  
mvar2.1976PL ................................................ 46  
mvar2.1982Muirhead .......................................... 48  
mvar2.2012ZXC ................................................ 49  
mvar2.LRT ....................................................... 50  
norm.1965SW .................................................. 51  
norm.1972SF .................................................. 52  
norm.1980JB .................................................. 54  
norm.1996AJB ................................................ 55  
norm.2008RJB ................................................ 56  
sim1.2017Liu .................................................. 57  
sim1.LRT ....................................................... 58  
sim2.2018HN .................................................. 59  
simplex.uniform ............................................... 60  
unif.2017YMq ................................................ 63  
unif.2017YMq ................................................ 63  
usek1d .......................................................... 64  
useknd .......................................................... 65  
var1.chisq .................................................... 66  
var2.F .......................................................... 68  
vark.1937Bartlett .............................................. 69
Description

Given a multivariate sample $X$ and hypothesized covariance matrix $\Sigma_0$, it tests

$$H_0 : \Sigma_x = \Sigma_0 \quad vs \quad H_1 : \Sigma_x \neq \Sigma_0$$

using the procedure by Fisher (2012). This method utilizes the generalized form of the inequality

$$\frac{1}{p} \sum_{i=1}^{p} (\lambda_i^r - 1)^{2s} \geq 0$$

and offers two types of test statistics $T_1$ and $T_2$ corresponding to the case $(r, s) = (1, 2)$ and $(2, 1)$ respectively.

Usage

cov1.2012Fisher(X, Sigma0 = diag(ncol(X)), type)

Arguments

- $X$ an $(n \times p)$ data matrix where each row is an observation.
- $\text{Sigma0}$ a $(p \times p)$ given covariance matrix.
- $\text{type}$ 1 or 2 for corresponding statistic from the paper.

Value

a (list) object of S3 class htest containing:

- $\text{statistic}$ a test statistic.
- $\text{p.value}$ $p$-value under $H_0$.
- $\text{alternative}$ alternative hypothesis.
- $\text{method}$ name of the test.
- $\text{data.name}$ name(s) of provided sample data.

References

Fisher TJ (2012). “On testing for an identity covariance matrix when the dimensionality equals or exceeds the sample size.” *Journal of Statistical Planning and Inference*, 142(1), 312–326. ISSN 03783758.
Examples

```r
## CRAN-purpose small example
cov1.2012Fisher(smallX) # run the test

## empirical Type 1 error
niter = 1000
counter1 = rep(0,niter) # p-values of the type 1
counter2 = rep(0,niter) # p-values of the type 2
for (i in 1:niter){
  X = matrix(rnorm(50*5), ncol=50) # (n,p) = (5,50)
  counter1[i] = ifelse(cov1.2012Fisher(X, type=1)$p.value < 0.05, 1, 0)
  counter2[i] = ifelse(cov1.2012Fisher(X, type=2)$p.value < 0.05, 1, 0)
}
## print the result
cat(paste("n
* Example for 'cov1.2012Fisher' n","n",
"* empirical error with statistic 1 : ", round(sum(counter1/niter),5),"n",
"* empirical error with statistic 2 : ", round(sum(counter2/niter),5),"n",sep=""))
```

---

**cov1.2015WL**  
*One-sample Test for Covariance Matrix by Wu and Li (2015)*

**Description**

Given a multivariate sample $X$ and hypothesized covariance matrix $\Sigma_0$, it tests

$$H_0: \Sigma_x = \Sigma_0 \quad vs \quad H_1: \Sigma_x \neq \Sigma_0$$

using the procedure by Wu and Li (2015). They proposed to use $m$ number of multiple random projections since only a single operation might attenuate the efficacy of the test.

**Usage**

`cov1.2015WL(X, Sigma0 = diag(ncol(X)), m = 25)`

**Arguments**

- `X` an $(n \times p)$ data matrix where each row is an observation.
- `Sigma0` a $(p \times p)$ given covariance matrix.
- `m` the number of random projections to be applied.
Value

a (list) object of S3 class htest containing:

statistic a test statistic.
p.value $p$-value under $H_0$.
alternative alternative hypothesis.
method name of the test.
data.name name(s) of provided sample data.

References


Examples

## CRAN-purpose small example

cov1.2015WL(smallX) # run the test

## empirical Type 1 error
## compare effects of m=5, 10, 50

niter = 1000
rec1 = rep(0,niter) # for m=5
rec2 = rep(0,niter) # m=10
rec3 = rep(0,niter) # m=50

for (i in 1:niter){
  X = matrix(rnorm(10*10), ncol=10) # (n,p) = (10,50)
  rec1[i] = ifelse(cov1.2015WL(X, m=5)$p.value < 0.05, 1, 0)
  rec2[i] = ifelse(cov1.2015WL(X, m=10)$p.value < 0.05, 1, 0)
  rec3[i] = ifelse(cov1.2015WL(X, m=50)$p.value < 0.05, 1, 0)
}

## print the result

cat(paste("* Example for "cov1.2015WL"Var" \\
"* Type 1 error with m=5 : ",round(sum(rec1/niter),5),"\n", 
"* Type 1 error with m=10 : ",round(sum(rec2/niter),5),"\n", 
"* Type 1 error with m=50 : ",round(sum(rec3/niter),5),"\n",sep="\"\")

---

Two-sample Test for High-Dimensional Covariances by Li and Chen (2012)
Description

Given two multivariate data \( X \) and \( Y \) of same dimension, it tests

\[
H_0 : \Sigma_x = \Sigma_y \quad \text{vs} \quad H_1 : \Sigma_x \neq \Sigma_y
\]

using the procedure by Li and Chen (2012).

Usage

\[
\text{cov2.2012LC}(X, Y, \text{use.unbiased} = \text{TRUE})
\]

Arguments

\( X \)  
 an \(( n_x \times p )\) data matrix of 1st sample.

\( Y \)  
 an \(( n_y \times p )\) data matrix of 2nd sample.

\( \text{use.unbiased} \)  
a logical; \text{TRUE} to use up to 4th-order U-statistics as proposed in the paper, \text{FALSE} for faster run under an assumption that \( \mu_h = 0 \) (default: \text{TRUE}).

Value

a (list) object of S3 class \text{htest} containing:

\text{statistic}  
a test statistic.

\text{p.value}  
p-value under \( H_0 \).

\text{alternative}  
alternative hypothesis.

\text{method}  
name of the test.

\text{data.name}  
name(s) of provided sample data.

References


Examples

```r
## CRAN-purpose small example
smallX = matrix(rnorm(10*4),ncol=5)
smallY = matrix(rnorm(10*4),ncol=5)
cov2.2012LC(smallX, smallY) # run the test

## Not run:
## empirical Type 1 error : use 'biased' version for faster computation
niter = 1000
counter = rep(0,niter)
for (i in 1:niter){
  X = matrix(rnorm(500*25), ncol=10)
  Y = matrix(rnorm(500*25), ncol=10)
  counter[i] = ifelse(cov2.2012LC(X,Y,use.unbiased=FALSE)$p.value < 0.05,1,0)
```
print(paste0("iteration ",i,"/1000 complete."))
}

## print the result
cat(paste("\n* Example for 'cov2.2012LC'\n","*
","* number of rejections : ",sum(counter),"\n",
","* total number of trials : ",niter,"\n",
","* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))

## End(Not run)

---

cov2.2013CLX Two-sample Test for Covariance Matrices by Cai, Liu, and Xia (2013)

**Description**

Given two multivariate data \(X\) and \(Y\) of same dimension, it tests

\[
H_0 : \Sigma_x = \Sigma_y \quad vs \quad H_1 : \Sigma_x \neq \Sigma_y
\]

using the procedure by Cai, Liu, and Xia (2013).

**Usage**

`cov2.2013CLX(X, Y)`

**Arguments**

- `X` an \((n_x \times p)\) data matrix of 1st sample.
- `Y` an \((n_y \times p)\) data matrix of 2nd sample.

**Value**

a (list) object of S3 class `htest` containing:

- `statistic` a test statistic.
- `p.value` \(p\)-value under \(H_0\).
- `alternative` alternative hypothesis.
- `method` name of the test.
- `data.name` name(s) of provided sample data.

**References**

Examples

```r
## CRAN-purpose small example
smallX = matrix(rnorm(10*3),ncol=3)
smallY = matrix(rnorm(10*3),ncol=3)
cov2.2013CLX(smallX, smallY) # run the test

## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
  X = matrix(rnorm(50*5), ncol=10)
  Y = matrix(rnorm(50*5), ncol=10)
  counter[i] = ifelse(cov2.2013CLX(X, Y)$p.value < 0.05, 1, 0)
}
## print the result
cat(paste("* Example for "cov2.2013CLX"Var","* 
  * number of rejections : ", sum(counter),"*",  
  * total number of trials : ", niter,"*",  
  * empirical Type 1 error : ",round(sum(counter/niter),5),"*",sep=""))
```

---

**cov2.2015WL**  
*Two-sample Test for Covariance Matrices by Wu and Li (2015)*

**Description**  
Given two multivariate data $X$ and $Y$ of same dimension, it tests  

$$H_0 : \Sigma_x = \Sigma_y \ vs \ H_1 : \Sigma_x \neq \Sigma_y$$

using the procedure by Wu and Li (2015).

**Usage**  
`cov2.2015WL(X, Y, m = 50)`

**Arguments**

- `X`  
  an $(n_x \times p)$ data matrix of 1st sample.
- `Y`  
  an $(n_y \times p)$ data matrix of 2nd sample.
- `m`  
  the number of random projections to be applied.
Value

a (list) object of S3 class htest containing:

- **statistic** a test statistic.
- **p.value** \( p \)-value under \( H_0 \).
- **alternative** alternative hypothesis.
- **method** name of the test.
- **data.name** name(s) of provided sample data.

References


Examples

```r
## CRAN-purpose small example
smallX = matrix(rnorm(10*3),ncol=3)
smallY = matrix(rnorm(10*3),ncol=3)
cov2.2015WL(smallX, smallY) # run the test

## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
  X = matrix(rnorm(50*5), ncol=10)
  Y = matrix(rnorm(50*5), ncol=10)
  counter[i] = ifelse(cov2.2015WL(X, Y)$p.value < 0.05, 1, 0)
}
## print the result
cat(paste("\n* Example for 'cov2.2015WL'\n","* number of rejections : ", sum(counter),"\n",
"* total number of trials : ", niter,"\n",
"* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
```
Description

Given univariate samples $X_1, \ldots, X_k$, it tests

$$H_0 : \Sigma_1 = \cdots = \Sigma_k \ vs \ H_1 : \text{at least one equality does not hold}$$

using the procedure by Schott (2001) using Wald statistics. In the original paper, it provides 4 different test statistics for general elliptical distribution cases. However, we only deliver the first one with an assumption of multivariate normal population.

Usage

covk.2001Schott(dlist)

Arguments

dlist \hspace{1em} \text{a list of length } k \text{ where each element is a sample matrix of same dimension.}

Value

a (list) object of S3 class htest containing:

- **statistic** a test statistic.
- **p.value** $p$-value under $H_0$.
- **alternative** alternative hypothesis.
- **method** name of the test.
- **data.name** name(s) of provided sample data.

References


Examples

```r
## CRAN-purpose small example
tinylist = list()
for (i in 1:3){ # consider 3-sample case
tinylist[[i]] = matrix(rnorm(10*3),ncol=3)
}
covk.2001Schott(tinylist) # run the test

## Not run:
## test when k=5 samples with (n,p) = (100,20)
## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
    mylist = list()
    for (j in 1:5){
        mylist[[j]] = matrix(rnorm(100*20),ncol=20)
    }
covk.2001Schott(mylist) # run the test
}
```
covk.2007Schott

Test for Homogeneity of Covariances by Schott (2007)

Description
Given univariate samples $X_1, \ldots, X_k$, it tests

$$H_0 : \Sigma_1 = \cdots = \Sigma_k \quad vs \quad H_1 : \text{at least one equality does not hold}$$

using the procedure by Schott (2007).

Usage
covk.2007Schott(dlist)

Arguments
dlist a list of length $k$ where each element is a sample matrix of same dimension.

Value
a (list) object of S3 class htest containing:

- statistic a test statistic.
- p.value $p$-value under $H_0$.
- alternative alternative hypothesis.
- method name of the test.
- data.name name(s) of provided sample data.

References
Schott JR (2007). “A test for the equality of covariance matrices when the dimension is large relative to the sample sizes.” *Computational Statistics & Data Analysis*, 51(12), 6535–6542. ISSN 01679473.
Examples

```r
## CRAN-purpose small example
tinylist = list()
for (i in 1:3){ # consider 3-sample case
  tinylist[[i]] = matrix(rnorm(10*3),ncol=3)
}
covk.2007Schott(tinylist) # run the test

## test when k=4 samples with (n,p) = (100,20)
## empirical Type 1 error
niter = 1234
counter = rep(0,niter) # record p-values
for (i in 1:niter){
  mylist = list()
  for (j in 1:4){
    mylist[[j]] = matrix(rnorm(100*20),ncol=20)
  }
  counter[i] = ifelse(covk.2007Schott(mylist)$p.value < 0.05, 1, 0)
}

## print the result
cat(paste("* Example for \"covk.2007Schott\"\n","* number of rejections : ", sum(counter),"\n",
  "* total number of trials : ", niter,"\n",
  "* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep="\")
```

Description

Given two samples (either univariate or multivariate) \( X \) and \( Y \) of same dimension, it tests

\[
H_0 : F_X = F_Y \quad \text{vs} \quad H_1 : F_X \neq F_Y
\]

using the procedure by Biswas and Ghosh (2014) in a nonparametric way based on pairwise distance measures. Both asymptotic and permutation-based determination of \( p \)-values are supported.

Usage

eqdist.2014BG(X, Y, method = c("permutation", "asymptotic"), nreps = 999)
Arguments

- **X**: a vector/matrix of 1st sample.
- **Y**: a vector/matrix of 2nd sample.
- **method**: method to compute $p$-value. Using initials is possible, "p" for permutation tests. Case insensitive.
- **nreps**: the number of permutations to be run when method="permutation".

Value

A (list) object of S3 class htest containing:

- **statistic**: a test statistic.
- **p.value**: $p$-value under $H_0$.
- **alternative**: alternative hypothesis.
- **method**: name of the test.
- **data.name**: name(s) of provided sample data.

References


Examples

```r
## CRAN-purpose small example
smallX = matrix(rnorm(10*3),ncol=3)
smallY = matrix(rnorm(10*3),ncol=3)
eqdist.2014BG(smallX, smallY) # run the test

## Not run:
## compare asymptotic and permutation-based powers
set.seed(777)
ntest = 1000
pval.a = rep(0,ntest)
pval.p = rep(0,ntest)
for (i in 1:ntest){
  x = matrix(rnorm(100), nrow=5)
  y = matrix(rnorm(100), nrow=5)
  pval.a[i] = ifelse(eqdist.2014BG(x,y,method="a")$p.value<0.05,1,0)
  pval.p[i] = ifelse(eqdist.2014BG(x,y,method="p",nreps=100)$p.value <0.05,1,0)
}
## print the result
cat(paste("\n* EMPIRICAL TYPE I ERROR COMPARISON \n","*\n","* Asymptotics : ", round(sum(pval.a/ntest),5),"\n",
","* Permutation : ", round(sum(pval.p/ntest),5),"\n",sep=""))
```
One-sample Hotelling’s T-squared Test for Multivariate Mean

Description
Given a multivariate sample $X$ and hypothesized mean $\mu_0$, it tests

$$H_0 : \mu_x = \mu_0 \ vs \ H_1 : \mu_x \neq \mu_0$$

using the procedure by Hotelling (1931).

Usage
mean1.1931Hotelling(X, mu0 = rep(0, ncol(X)))

Arguments

- $X$ an $(n \times p)$ data matrix where each row is an observation.
- $\mu_0$ a length-$p$ mean vector of interest.

Value

a (list) object of S3 class htest containing:

- statistic a test statistic.
- p.value $p$-value under $H_0$.
- alternative alternative hypothesis.
- method name of the test.
- data.name name(s) of provided sample data.

References

Examples

```r
## CRAN-purpose small example
smallX = matrix(rnorm(10*3),ncol=3)
mean1.1931Hotelling(smallX) # run the test
```

```r
## Not run:
## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
```
for (i in 1:niter){
    X = matrix(rnorm(50*5), ncol=5)
    counter[i] = ifelse(mean1.1931Hotelling(X)$p.value < 0.05, 1, 0)
}

## print the result
cat(paste("\n* Example for 'mean1.1931Hotelling'\n","\n","* number of rejections : ", sum(counter),"\n","* total number of trials : ", niter,"\n","* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))

## End(Not run)

mean1.1958Dempster  One-sample Test for Mean Vector by Dempster (1958, 1960)

Description

Given a multivariate sample $X$ and hypothesized mean $\mu_0$, it tests

$H_0 : \mu_x = \mu_0$ vs $H_1 : \mu_x \neq \mu_0$

using the procedure by Dempster (1958, 1960).

Usage

mean1.1958Dempster(X, mu0 = rep(0, ncol(X)))

Arguments

- **X** an $(n \times p)$ data matrix where each row is an observation.
- **mu0** a length-$p$ mean vector of interest.

Value

a (list) object of S3 class htest containing:

- **statistic** a test statistic.
- **p.value** $p$-value under $H_0$.
- **alternative** alternative hypothesis.
- **method** name of the test.
- **data.name** name(s) of provided sample data.

References


Dempster AP (1960). “A Significance Test for the Separation of Two Highly Multivariate Small Samples.” *Biometrics*, 16(1), 41. ISSN 0006341X.
Examples

```r
## CRAN-purpose small example
smallX = matrix(rnorm(10*3),ncol=3)
mean1.1958Dempster(smallX) # run the test

## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
  X = matrix(rnorm(50*5),ncol=50)
  counter[i] = ifelse(mean1.1958Dempster(X)$p.value < 0.05, 1, 0)
}
## print the result
cat(paste("\n* Example for 'mean1.1958Dempster'\n","\n","* number of rejections : ", sum(counter),"\n","* total number of trials : ", niter,"\n","* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
```

---

**mean1.1996BS**  
*One-sample Test for Mean Vector by Bai and Saranadasa (1996)*

**Description**

Given a multivariate sample $X$ and hypothesized mean $\mu_0$, it tests

$$H_0: \mu_x = \mu_0 \ vs \ H_1: \mu_x \neq \mu_0$$

using the procedure by Bai and Saranadasa (1996).

**Usage**

```r
mean1.1996BS(X, mu0 = rep(0, ncol(X)))
```

**Arguments**

- `X` an $(n \times p)$ data matrix where each row is an observation.
- `mu0` a length-$p$ mean vector of interest.

**Value**

a (list) object of S3 class `htest` containing:

- `statistic` a test statistic.
- `p.value` $p$-value under $H_0$.
- `alternative` alternative hypothesis.
- `method` name of the test.
- `data.name` name(s) of provided sample data.
References


Examples

```r
## CRAN-purpose small example
smallX = matrix(rnorm(10*3),ncol=3)
mean1.1996BS(smallX) # run the test

## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
  X = matrix(rnorm(50*5), ncol=25)
  counter[i] = ifelse(mean1.1996BS(X)$p.value < 0.05, 1, 0)
}

## print the result
cat(paste("\n* Example for 'mean1.1996BS'\n","*
","* number of rejections : ", sum(counter),"\n",
","* total number of trials : ", niter,"\n",
","* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
```

---

**One-sample Test for Mean Vector by Srivastava and Du (2008)**

**Description**

Given a multivariate sample $X$ and hypothesized mean $\mu_0$, it tests $H_0 : \mu_x = \mu_0$ vs $H_1 : \mu_x \neq \mu_0$

using the procedure by Srivastava and Du (2008).

**Usage**

```r
mean1.2008SD(X, mu0 = rep(0, ncol(X)))
```

**Arguments**

- **X**
  - an $(n \times p)$ data matrix where each row is an observation.
- **mu0**
  - a length-$p$ mean vector of interest.
Value

a (list) object of S3 class htest containing:

- **statistic** a test statistic.
- **p.value** p-value under $H_0$.
- **alternative** alternative hypothesis.
- **method** name of the test.
- **data.name** name(s) of provided sample data.

References


Examples

```r
## CRAN-purpose small example
smallX = matrix(rnorm(10*3), ncol=3)
mean1.2008SD(smallX) # run the test

## empirical Type 1 error
niter = 1000
counter = rep(0, niter) # record p-values
for (i in 1:niter){
  X = matrix(rnorm(50*5), ncol=5)
  counter[i] = ifelse(mean1.2008SD(X)$p.value < 0.05, 1, 0)
}
## print the result
cat(paste("\n* Example for 'mean1.2008SD' \n","\n","\n","\n","\n","\n","\n",
"* number of rejections : ", sum(counter),"\n",
"* total number of trials : ", niter, "\n",
"* empirical Type 1 error : ", round(sum(counter/niter),5),"\n", sep=""))
```

---

**mean1.ttest**  
*One-sample Student’s t-test for Univariate Mean*

### Description

Given an univariate sample $x$, it tests $H_0: \mu_x = \mu_0$ vs $H_1: \mu_x \neq \mu_0$ using the procedure by Student (1908).
mean1.ttest

Usage

mean1.ttest(x, mu0 = 0, alternative = c("two.sided", "less", "greater"))

Arguments

x
a length-\(n\) data vector.

mu0
hypothesized mean \(\mu_0\).

alternative
specifying the alternative hypothesis.

Value

a (list) object of S3 class htest containing:

statistic
a test statistic.

p.value
\(p\)-value under \(H_0\).

alternative
alternative hypothesis.

method
name of the test.

data.name
name(s) of provided sample data.

References

Student (1908). “The Probable Error of a Mean.” Biometrika, 6(1), 1. ISSN 00063444.


Examples

```r
## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
  x = rnorm(10) # sample from N(0,1)
  counter[i] = ifelse(mean1.ttest(x)$p.value < 0.05, 1, 0)
}

## print the result
cat(paste("\n* Example for 'mean1.ttest'\n","\n","* number of rejections : ", sum(counter),"\n","* total number of trials : ", niter,"\n","* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
```
mean2.1931Hotelling  Two-sample Hotelling’s T-squared Test for Multivariate Means

Description
Given two multivariate data $X$ and $Y$ of same dimension, it tests

$H_0 : \mu_x = \mu_y \; \text{vs} \; H_1 : \mu_x \neq \mu_y$

using the procedure by Hotelling (1931).

Usage
mean2.1931Hotelling(X, Y, paired = FALSE, var.equal = TRUE)

Arguments
X an $(n_x \times p)$ data matrix of 1st sample.
Y an $(n_y \times p)$ data matrix of 2nd sample.
paired a logical; whether you want a paired Hotelling’s test.
var.equal a logical; whether to treat the two covariances as being equal.

Value
a (list) object of S3 class htest containing:

statistic a test statistic.
p.value $p$-value under $H_0$.
alternative alternative hypothesis.
method name of the test.
data.name name(s) of provided sample data.

References

Examples
## CRAN-purpose small example
smallX = matrix(rnorm(10*3),ncol=3)
smallY = matrix(rnorm(10*3),ncol=3)
mean2.1931Hotelling(smallX, smallY) # run the test

## generate two samples from standard normal distributions.
X = matrix(rnorm(50*5), ncol=5)
Y = matrix(rnorm(77*5), ncol=5)

## run single test
print(mean2.1931Hotelling(X,Y))

## empirical Type 1 error
niter = 1000
counter = rep(0, niter)  # record p-values
for (i in 1:niter){
  X = matrix(rnorm(50*5), ncol=5)
  Y = matrix(rnorm(77*5), ncol=5)

  counter[i] = ifelse(mean2.1931Hotelling(X,Y)$p.value < 0.05, 1, 0)
}

## print the result
cat(paste("\n* Example for 'mean2.1931Hotelling'\n","*
","* number of rejections : ", sum(counter),"\n",
","* total number of trials : ", niter,"\n",
","* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))

----------

**mean2.1958Dempster**  Two-sample Test for High-Dimensional Means by Dempster (1958, 1960)

**Description**

Given two multivariate data \(X\) and \(Y\) of same dimension, it tests

\[
H_0: \mu_x = \mu_y \quad \text{vs} \quad H_1: \mu_x \neq \mu_y
\]

using the procedure by Dempster (1958, 1960).

**Usage**

mean2.1958Dempster(X, Y)

**Arguments**

- **X**: an \((n_x \times p)\) data matrix of 1st sample.
- **Y**: an \((n_y \times p)\) data matrix of 2nd sample.

**Value**

A (list) object of S3 class htest containing:

- **statistic**: a test statistic.
p.value  $p$-value under $H_0$.
alternative  alternative hypothesis.
method  name of the test.
data.name  name(s) of provided sample data.

References


Dempster AP (1960). “A Significance Test for the Separation of Two Highly Multivariate Small Samples.” *Biometrics*, **16**(1), 41. ISSN 0006341X.

Examples

```r
## CRAN-purpose small example
smallX = matrix(rnorm(10*3),ncol=3)
smallY = matrix(rnorm(10*3),ncol=3)
mean2.1958Dempster(smallX, smallY) # run the test

## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
  X = matrix(rnorm(50*5), ncol=10)
  Y = matrix(rnorm(50*5), ncol=10)
  counter[i] = ifelse(mean2.1958Dempster(X,Y)$p.value < 0.05, 1, 0)
}
## print the result
cat(paste("* Example for 'mean2.1958Dempster'\n","* number of rejections : ", sum(counter),"\n","* total number of trials : ", niter,"\n","* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
```

---

**mean2.1965Yao**  
*Two-sample Test for Multivariate Means by Yao (1965)*

**Description**

Given two multivariate data $X$ and $Y$ of same dimension, it tests

$$H_0 : \mu_x = \mu_y \text{ vs } H_1 : \mu_x \neq \mu_y$$

using the procedure by Yao (1965) via multivariate modification of Welch’s approximation of degrees of freedoms.
Usage

mean2.1965Yao(X, Y)

Arguments

X an \((n_x \times p)\) data matrix of 1st sample.
Y an \((n_y \times p)\) data matrix of 2nd sample.

Value

a (list) object of S3 class htest containing:

- statistic a test statistic.
- p.value \(p\)-value under \(H_0\).
- alternative alternative hypothesis.
- method name of the test.
- data.name name(s) of provided sample data.

References


Examples

```r
## CRAN-purpose small example
smallX = matrix(rnorm(10*3),ncol=3)
smallY = matrix(rnorm(10*3),ncol=3)
mean2.1965Yao(smallX, smallY) # run the test
```

```r
## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
  X = matrix(rnorm(50*5), ncol=10)
  Y = matrix(rnorm(50*5), ncol=10)

  counter[i] = ifelse(mean2.1965Yao(X,Y)$p.value < 0.05, 1, 0)
}
```

```r
## print the result
cat(paste("\n* Example for 'mean2.1965Yao'\n","*
","* number of rejections : ", sum(counter),"\n",
","* total number of trials : ", niter,"\n",
","* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
```
Description

Given two multivariate data $X$ and $Y$ of same dimension, it tests

$$H_0 : \mu_x = \mu_y \text{ vs } H_1 : \mu_x \neq \mu_y$$

using the procedure by Johansen (1980) by adapting Welch-James approximation of the degree of freedom for Hotelling’s $T^2$ test.

Usage

`mean2.1980Johansen(X, Y)`

Arguments

- `X`: an $(n_x \times p)$ data matrix of 1st sample.
- `Y`: an $(n_y \times p)$ data matrix of 2nd sample.

Value

A (list) object of S3 class `htest` containing:

- `statistic`: a test statistic.
- `p.value`: p-value under $H_0$.
- `alternative`: alternative hypothesis.
- `method`: name of the test.
- `data.name`: name(s) of provided sample data.

References


Examples

```r
## CRAN-purpose small example
smallX = matrix(rnorm(10*3),ncol=3)
smallY = matrix(rnorm(10*3),ncol=3)
mean2.1980Johansen(smallX, smallY) # run the test

## Not run:
## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
```
mean2.1986NVM

Two-sample Test for Multivariate Means by Nel and Van der Merwe (1986)

Description

Given two multivariate data $X$ and $Y$ of same dimension, it tests

$$H_0: \mu_x = \mu_y \text{ vs } H_1: \mu_x \neq \mu_y$$

using the procedure by Nel and Van der Merwe (1986).

Usage

mean2.1986NVM(X, Y)

Arguments

- **X**: an $$(n_x \times p)$$ data matrix of 1st sample.
- **Y**: an $$(n_y \times p)$$ data matrix of 2nd sample.

Value

a (list) object of S3 class `htest` containing:

- **statistic**: a test statistic.
- **p.value**: $p$-value under $H_0$.
- **alternative**: alternative hypothesis.
- **method**: name of the test.
- **data.name**: name(s) of provided sample data.
References


Examples

```r
## CRAN-purpose small example
smallX = matrix(rnorm(10*3),ncol=3)
smallY = matrix(rnorm(10*3),ncol=3)
mean2.1986NVM(smallX, smallY) # run the test

## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
    X = matrix(rnorm(50*5), ncol=10)
    Y = matrix(rnorm(50*5), ncol=10)
    counter[i] = ifelse(mean2.1986NVM(X,Y)$p.value < 0.05, 1, 0)
}
## print the result
cat(paste("\n* Example for \"Var mean2.1986NVM\"\n","\n* number of rejections : ", sum(counter),"\n",
* total number of trials : ", niter,"\n",
* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
```

mean2.1996BS

*Two-sample Test for High-Dimensional Means by Bai and Saranadasa (1996)*

Description

Given two multivariate data $X$ and $Y$ of same dimension, it tests

$H_0: \mu_x = \mu_y \hspace{1cm} vs \hspace{1cm} H_1: \mu_x \neq \mu_y$

using the procedure by Bai and Saranadasa (1996).

Usage

```r
mean2.1996BS(X, Y)
```
Arguments

X  an \((n_x \times p)\) data matrix of 1st sample.
Y  an \((n_y \times p)\) data matrix of 2nd sample.

Value

a (list) object of S3 class htest containing:

- **statistic**  a test statistic.
- **p.value**  \(p\)-value under \(H_0\).
- **alternative**  alternative hypothesis.
- **method**  name of the test.
- **data.name**  name(s) of provided sample data.

References


Examples

```r
## CRAN-purpose small example
smallX = matrix(rnorm(10*3),ncol=3)
smallY = matrix(rnorm(10*3),ncol=3)
mean2.1996BS(smallX, smallY) # run the test

## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
  X = matrix(rnorm(50*5), ncol=10)
  Y = matrix(rnorm(50*5), ncol=10)
  counter[i] = ifelse(mean2.1996BS(X,Y)$p.value < 0.05, 1, 0)
}
## print the result
cat(paste("\n* number of rejections : ", sum(counter),"\n","\n* total number of trials : ", niter,"\n","\n* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
```
Two-sample Test for Multivariate Means by Krishnamoorthy and Yu (2004)

Description
Given two multivariate data X and Y of same dimension, it tests

\[ H_0 : \mu_x = \mu_y \text{ vs } H_1 : \mu_x \neq \mu_y \]

using the procedure by Krishnamoorthy and Yu (2004), which is a modified version of Nel and Van der Merwe (1986).

Usage
mean2.2004KY(X, Y)

Arguments
X an \((n_x \times p)\) data matrix of 1st sample.
Y an \((n_y \times p)\) data matrix of 2nd sample.

Value
a (list) object of S3 class htest containing:

- **statistic** a test statistic.
- **p.value** \(p\)-value under \(H_0\).
- **alternative** alternative hypothesis.
- **method** name of the test.
- **data.name** name(s) of provided sample data.

References

Examples
```r
## CRAN-purpose small example
smallX = matrix(rnorm(10*3),ncol=3)
smallY = matrix(rnorm(10*3),ncol=3)
mean2.2004KY(smallX, smallY) # run the test

## Not run:
## empirical Type I error
niter = 1000
counter = rep(0,niter) # record p-values
```
for (i in 1:niter){
    X = matrix(rnorm(50*5), ncol=10)
    Y = matrix(rnorm(50*5), ncol=10)
    counter[i] = ifelse(mean2.2004KY(X,Y)$p.value < 0.05, 1, 0)
}

## print the result
cat(paste("\n\n* Example for \'mean2.2004KY\' \n","\n",
"* number of rejections : ", sum(counter),"\n",
"* total number of trials : ", niter,"\n",
"* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))

## End(Not run)

mean2.2008SD

Two-sample Test for High-Dimensional Means by Srivastava and Du (2008)

Description

Given two multivariate data X and Y of same dimension, it tests

\[ H_0 : \mu_x = \mu_y \] vs \[ H_1 : \mu_x \neq \mu_y \]

using the procedure by Srivastava and Du (2008).

Usage

mean2.2008SD(X, Y)

Arguments

X

an \( (n_x \times p) \) data matrix of 1st sample.

Y

an \( (n_y \times p) \) data matrix of 2nd sample.

Value

a (list) object of S3 class htest containing:

statistic  a test statistic.

p.value  \( p \)-value under \( H_0 \).

alternative  alternative hypothesis.

method  name of the test.

data.name  name(s) of provided sample data.
References


Examples

```r
## CRAN-purpose small example
smallX = matrix(rnorm(10*3),ncol=3)
smallY = matrix(rnorm(10*3),ncol=3)
mean2.2008SD(smallX, smallY) # run the test

## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
  X = matrix(rnorm(50*5), ncol=10)
  Y = matrix(rnorm(50*5), ncol=10)
  counter[i] = ifelse(mean2.2008SD(X,Y)$p.value < 0.05, 1, 0)
}

## print the result
cat(paste("\n* Example for 'mean2.2008SD'\n","*
  * number of rejections : ", sum(counter),"\n",
  * total number of trials : ", niter,"\n",
  * empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
```

---

**mean2.2011LJW**

*Two-sample Test for Multivariate Means by Lopes, Jacob, and Wainwright (2011)*

**Description**

Given two multivariate data $X$ and $Y$ of same dimension, it tests

$$H_0: \mu_X = \mu_Y \quad vs \quad H_1: \mu_X \neq \mu_Y$$

using the procedure by Lopes, Jacob, and Wainwright (2011) using random projection. Due to solving system of linear equations, we suggest you to opt for asymptotic-based $p$-value computation unless truly necessary for random permutation tests.

**Usage**

```r
mean2.2011LJW(X, Y, method = c("asymptotic", "MC"), nreps = 1000)
```
Arguments

\(X\) an \((n_x \times p)\) data matrix of 1st sample.

\(Y\) an \((n_y \times p)\) data matrix of 2nd sample.

\textit{method}\ method to compute \(p\)-value. "asymptotic" for using approximating null distribution, and "MC" for random permutation tests. Using initials is possible, "a" for asymptotic for example.

\textit{nreps}\ the number of permutation iterations to be run when \textit{method}="MC".

Value

a (list) object of S3 class \texttt{htest} containing:

\texttt{statistic} a test statistic.

\texttt{p.value} \(p\)-value under \(H_0\).

\texttt{alternative} alternative hypothesis.

\texttt{method} name of the test.

\texttt{data.name} name(s) of provided sample data.

References


Examples

```r
## CRAN-purpose small example
smallX = matrix(rnorm(10*3),ncol=10)
smallY = matrix(rnorm(10*3),ncol=10)
mean2.2011LJW(smallX, smallY) # run the test
```

```r
## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
  X = matrix(rnorm(10*20), ncol=20)
  Y = matrix(rnorm(10*20), ncol=20)
  counter[i] = ifelse(mean2.2011LJW(X,Y)$p.value < 0.05, 1, 0)
}
```

```r
## print the result
cat(paste("\nExample for 'mean2.2011LJW'\n","\n","\n","\n","\n","\n","\n","\n","\n","\n",
"* number of rejections : ", sum(counter),"\n",
"* total number of trials : ", niter,"\n",
"* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
```
Description

Given two multivariate data $X$ and $Y$ of same dimension, it tests

$$H_0: \mu_x = \mu_y \quad \text{vs} \quad H_1: \mu_x \neq \mu_y$$

using the procedure by Cai, Liu, and Xia (2014) which is equivalent to test

$$H_0: \Omega(\mu_x - \mu_y) = 0$$

for an inverse covariance (or precision) $\Omega$. When $\Omega$ is not given and known to be sparse, it is first estimated with CLIME estimator. Otherwise, adaptive thresholding estimator is used. Also, if two samples are assumed to have different covariance structure, it uses weighting scheme for adjustment.

Usage

```r
mean2.2014CLX( 
  X, 
  Y, 
  precision = c("sparse", "unknown"), 
  delta = 2, 
  Omega = NULL, 
  cov.equal = TRUE 
)
```

Arguments

- **X**: an $(n_x \times p)$ data matrix of 1st sample.
- **Y**: an $(n_y \times p)$ data matrix of 2nd sample.
- **precision**: type of assumption for a precision matrix (default: "sparse").
- **delta**: an algorithmic parameter for adaptive thresholding estimation (default: 2).
- **Omega**: precision matrix; if NULL, an estimate is used. Otherwise, a $(p \times p)$ inverse covariance should be provided.
- **cov.equal**: a logical to determine homogeneous covariance assumption.

Value

a (list) object of S3 class `htest` containing:

- **statistic**: a test statistic.
- **p.value**: $p$-value under $H_0$.
- **alternative**: alternative hypothesis.
- **method**: name of the test.
- **data.name**: name(s) of provided sample data.
References


Examples

```r
## CRAN-purpose small example
smallX = matrix(rnorm(10*3), ncol=3)
smallY = matrix(rnorm(10*3), ncol=3)
mean2.2014CLX(smallX, smallY, precision="unknown")
mean2.2014CLX(smallX, smallY, precision="sparse")

## Not run:
## empirical Type 1 error
niter = 100
counter = rep(0, niter) # record p-values
for (i in 1:niter){
  X = matrix(rnorm(50*5), ncol=10)
  Y = matrix(rnorm(50*5), ncol=10)
  counter[i] = ifelse(mean2.2014CLX(X, Y)$p.value < 0.05, 1, 0)
}

## print the result
cat(paste("\n* Example for 'mean2.2014CLX'
","*
","* number of rejections : ", sum(counter),"\n",
","* total number of trials : ", niter,"\n",
","* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))

## End(Not run)
```

Description

Given two multivariate data $X$ and $Y$ of same dimension, it tests

$$H_0 : \mu_x = \mu_y \quad vs \quad H_1 : \mu_x \neq \mu_y$$

using the procedure by Thulin (2014) using random subspace methods. We did not enable parallel computing schemes for this in that it might incur huge computational burden since it entirely depends on random permutation scheme.

Usage

```r
mean2.2014Thulin(X, Y, B = 100, nreps = 1000)
```
Arguments

X an \((n_x \times p)\) data matrix of 1st sample.
Y an \((n_y \times p)\) data matrix of 2nd sample.
B the number of selected subsets for averaging. \(B \geq 100\) is recommended.
nreps the number of permutation iterations to be run.

Value

a (list) object of S3 class htest containing:

- **statistic** a test statistic.
- **p.value** \(p\)-value under \(H_0\).
- **alternative** alternative hypothesis.
- **method** name of the test.
- **data.name** name(s) of provided sample data.

References


Examples

```r
## CRAN-purpose small example
smallX = matrix(rnorm(10*3), ncol=10)
smallY = matrix(rnorm(10*3), ncol=10)
mean2.2014Thulin(smallX, smallY, B=10, nreps=10) # run the test

## Compare with 'mean2.2011LJW'
## which is based on random projection.
n = 33  # number of observations for each sample
p = 100  # dimensionality
X = matrix(rnorm(n*p), ncol=p)
Y = matrix(rnorm(n*p), ncol=p)
mean2.2011LJW(X, Y, nreps=100, method="m")  # 2011LJW requires 'm' to be set.
mean2.2014Thulin(X, Y, nreps=100)
```
Two-sample Mean Test with Maximum Pairwise Bayes Factor

Description

Not Written Here - No Reference Yet.

Usage

mean2.mxPBF(X, Y, a0 = 0, b0 = 0, gamma = 1, nthreads = 1)

Arguments

X an \((n_x \times p)\) data matrix of 1st sample.
Y an \((n_y \times p)\) data matrix of 2nd sample.
a0 shape parameter for inverse-gamma prior (default: 0).
b0 scale parameter for inverse-gamma prior (default: 0).
gamma non-negative variance scaling parameter (default: 1).
nthreads number of threads for parallel execution via OpenMP (default: 1).

Value

a (list) object of S3 class htest containing:

- statistic maximum of pairwise Bayes factor.
- alternative alternative hypothesis.
- method name of the test.
- data.name name(s) of provided sample data.
- log.BF.vec vector of pairwise Bayes factors in natural log.

Examples

```r
## Not run:
## empirical Type 1 error with BF threshold = 10
niter = 1000
counter = rep(0, niter) # record p-values
for (i in 1:niter){
  X = matrix(rnorm(100*10), ncol=10)
  Y = matrix(rnorm(200*10), ncol=10)
  counter[i] = ifelse(mean2.mxPBF(X,Y)$statistic > 10, 1, 0)
}

## print the result
cat(paste("\n* Example for 'mean2.mxPBF'\n","*\n","* number of rejections : ", sum(counter),"\n"))
```
mean2.ttest

Two-sample Student’s t-test for Univariate Means

Description

Given two univariate samples \( x \) and \( y \), it tests

\[
H_0 : \mu^2_x \{\geq, \leq\} \mu^2_y \quad \text{vs} \quad H_1 : \mu^2_x \{\neq, <, >\} \mu^2_y
\]

using the procedure by Student (1908) and Welch (1947).

Usage

\[
\text{mean2.ttest(}
\quad x,
\quad y,
\quad \text{alternative} = \text{c("two.sided", "less", "greater")},
\quad \text{paired} = \FALSE,
\quad \text{var.equal} = \FALSE
\)\]

Arguments

- \( x \) \quad a length-\( n \) data vector.
- \( y \) \quad a length-\( m \) data vector.
- \( \text{alternative} \) \quad specifying the alternative hypothesis.
- \( \text{paired} \) \quad a logical; whether consider two samples as paired.
- \( \text{var.equal} \) \quad a logical; if \( \FALSE \), use Welch’s correction.

Value

a (list) object of S3 class \texttt{htest} containing:

- \texttt{statistic} \quad a test statistic.
- \texttt{p.value} \quad \( p \)-value under \( H_0 \).
- \texttt{alternative} \quad alternative hypothesis.
- \texttt{method} \quad name of the test.
- \texttt{data.name} \quad name(s) of provided sample data.
meank.2007Schott

References
Welch BL (1947). “The Generalization of ‘Student’s’ Problem when Several Different Population Variances are Involved.” *Biometrika*, 34(1/2), 28. ISSN 00063444.

Examples

```r
## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
  x = rnorm(57) # sample x from N(0,1)
  y = rnorm(89) # sample y from N(0,1)
  counter[i] = ifelse(mean2.ttest(x,y)$p.value < 0.05, 1, 0)
}
## print the result
cat(paste("\nExample for ‘mean2.ttest’\n","\n","number of rejections : ", sum(counter),"\n","total number of trials : ", niter,"\n","empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
```

---

meank.2007Schott

Test for Equality of Means by Schott (2007)

Description

Given univariate samples $X_1, \ldots, X_k$, it tests

$$H_0 : \mu_1 = \cdots \mu_k \quad vs \quad H_1 : \text{at least one equality does not hold}$$

using the procedure by Schott (2007). It can be considered as a generalization of two-sample testing procedure proposed by Bai and Saranadasa (1996).

Usage

meank.2007Schott(dlist)

Arguments

dlist a list of length $k$ where each element is a sample matrix of same dimension.
Value

a (list) object of S3 class htest containing:

statistic a test statistic.
p.value $p$-value under $H_0$.
alternative alternative hypothesis.
method name of the test.
data.name name(s) of provided sample data.

References

Schott JR (2007). “Some high-dimensional tests for a one-way MANOVA.” *Journal of Multivariate Analysis*, 98(9), 1825–1839. ISSN 0047259X.

Examples

```r
## CRAN-purpose small example
tinylist = list()
for (i in 1:3){ # consider 3-sample case
tinylist[[i]] = matrix(rnorm(10*3),ncol=3)
}
meank.2007Schott(tinylist)

## test when k=5 samples with (n,p) = (10,50)
## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
    mylist = list()
    for (j in 1:5){
        mylist[[j]] = matrix(rnorm(10*5),ncol=5)
    }
    counter[i] = ifelse(meank.2007Schott(mylist)$p.value < 0.05, 1, 0)
}

counter[1] = ifelse(meank.2007Schott(mylist)$p.value < 0.05, 1, 0)
}
# print the result
cat(paste("\n* Example for 'meank.2007Schott'\n","* number of rejections : ", sum(counter),"\n",
"* total number of trials : ", niter,"\n",
"* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
```
Description

Given univariate samples $X_1, \ldots, X_k$, it tests

$$H_0 : \mu_1 = \cdots = \mu_k \quad \text{vs} \quad H_1 : \text{at least one equality does not hold}$$

using the procedure by Zhang and Xu (2009) by applying multivariate extension of Scheffe’s method of transformation.

Usage

```r
meank.2009ZX(dlist, method = c("L", "T"))
```

Arguments

dlist a list of length $k$ where each element is a sample matrix of same dimension.

method a method to be applied for the transformed problem. "L" for $L^2$-norm based method, and "T" for Hotelling’s test, which might fail due to dimensionality. Case insensitive.

Value

a (list) object of S3 class `htest` containing:

- `statistic` a test statistic.
- `p.value` $p$-value under $H_0$.
- `alternative` alternative hypothesis.
- `method` name of the test.
- `data.name` name(s) of provided sample data.

References


Examples

```r
## CRAN-purpose small example
tinylist = list()
for (i in 1:3){ # consider 3-sample case
tinylist[[i]] = matrix(rnorm(10*3),ncol=3)
}
meank.2009ZX(tinylist) # run the test
```
## test when k=5 samples with (n,p) = (100,20)
## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
  mylist = list()
  for (j in 1:5){
    mylist[[j]] = matrix(rnorm(100*10),ncol=10)
  }
  counter[i] = ifelse(meank.2009ZX(mylist, method="L")$p.value < 0.05, 1, 0)
}
## print the result
cat(paste("\n* Example for 'meank.2009ZX'\n","*
","* number of rejections : ", sum(counter),"\n",
","* total number of trials : ", niter,"\n",
","* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))

---

**meank.2019CPH**

**Test for Equality of Means by Cao, Park, and He (2019)**

**Description**

Given univariate samples $X_1, \ldots, X_k$, it tests

$$H_0 : \mu_1 = \cdots = \mu_k \ \text{vs} \ \ H_1 : \text{at least one equality does not hold}$$

using the procedure by Cao, Park, and He (2019).

**Usage**

```r
meank.2019CPH(dlist, method = c("original", "Hu"))
```

**Arguments**

- **dlist**: a list of length $k$ where each element is a sample matrix of same dimension.
- **method**: a method to be applied to estimate variance parameter. "original" for the estimator proposed in the paper, and "Hu" for the one used in 2017 paper by Hu et al. Case insensitive and initials can be used as well.
Value

a (list) object of S3 class htest containing:

- **statistic** a test statistic.
- **p.value** $p$-value under $H_0$.
- **alternative** alternative hypothesis.
- **method** name of the test.
- **data.name** name(s) of provided sample data.

References


Examples

```r
## CRAN-purpose small example
tinylist = list()
for (i in 1:3){ # consider 3-sample case
tinylist[[i]] = matrix(rnorm(10*3),ncol=3)
}
meank.2019CPH(tinylist, method="o") # newly-proposed variance estimator
meank.2019CPH(tinylist, method="h") # adopt one from 2017Hu

## Not run:
## test when k=5 samples with (n,p) = (10,50)
## empirical Type 1 error
niter  = 10000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
  mylist = list()
  for (j in 1:5){
    mylist[[j]] = matrix(rnorm(10*50),ncol=50)
  }
  counter[i] = ifelse(meank.2019CPH(mylist)$p.value < 0.05, 1, 0)
}
## print the result
cat(paste("\n* Example for 'meank.2019CPH'\n","\n","* number of rejections : ", sum(counter),"\n",
","* total number of trials : ", niter,"\n",
","* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
```

## End(Not run)
Analysis of Variance for Equality of Means

Description

Given univariate samples \(X_1, \ldots, X_k\), it tests

\[ H_0 : \mu_1^2 = \cdots = \mu_k^2 \quad vs \quad H_1 : \text{at least one equality does not hold.} \]

Usage

\[
\text{meank.anova} (dlist)
\]

Arguments

\[
dlist \quad \text{a list of length } k \text{ where each element is a sample vector.}
\]

Value

a (list) object of S3 class htest containing:

- **statistic**: a test statistic.
- **p.value**: \(p\)-value under \(H_0\).
- **alternative**: alternative hypothesis.
- **method**: name of the test.
- **data.name**: name(s) of provided sample data.

Examples

```
## test when k=5 (samples)
## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
  mylist = list()
  for (j in 1:5){
    mylist[[j]] = rnorm(50)
  }
  counter[i] = ifelse(meank.anova(mylist)$p.value < 0.05, 1, 0)
}

## print the result
paste("\n* Example for 'meank.anova'\n","*
","* number of rejections : ", sum(counter),"\n",
", "* total number of trials : ", niter,"\n",
", "* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
```
Description

Given two univariate samples \( x \) and \( y \), it tests

\[
H_0 : \mu_x = \mu_0, \sigma^2_x = \sigma_0^2 \quad vs 
H_1 : \text{not } H_0
\]

using asymptotic likelihood ratio test.

Usage

\texttt{mvar1.1998AS(x, mu0 = 0, var0 = 1)}

Arguments

- \texttt{x} a length-\( n \) data vector.
- \texttt{mu0} hypothesized mean \( \mu_0 \).
- \texttt{var0} hypothesized variance \( \sigma_0^2 \).

Value

a (list) object of S3 class \texttt{htest} containing:

- \texttt{statistic} a test statistic.
- \texttt{p.value} \( p \)-value under \( H_0 \).
- \texttt{alternative} alternative hypothesis.
- \texttt{method} name of the test.
- \texttt{data.name} name(s) of provided sample data.

References


Examples

```r
## CRAN-purpose small example
mvar1.1998AS(rnorm(10))
```

```r
## Not run:
## empirical Type 1 error
niter = 1000
counter = rep(0,niter)  # record p-values
for (i in 1:niter){
```
x = rnorm(100)  # sample x from N(0,1)

counter[i] = ifelse(mvar1.1998AS(x)$p.value < 0.05, 1, 0)
}

## print the result
cat(paste("\nExample for 'mvar1.1998AS'\n", "\n\n* number of rejections : ", sum(counter), "\n\n* total number of trials : ", niter, "\n\n* empirical Type 1 error : ", round(sum(counter/niter),5), "\n\n", sep=""))

## End(Not run)

---

mvar1.LRT

One-sample Simultaneous Likelihood Ratio Test of Mean and Variance

**Description**

Given two univariate samples $x$ and $y$, it tests

$$H_0 : \mu_x = \mu_0, \sigma^2_x = \sigma^2_0 \quad vs \quad H_1 : \text{not } H_0$$

using likelihood ratio test.

**Usage**

mvar1.LRT(x, mu0 = 0, var0 = 1)

**Arguments**

- **x**: a length-$n$ data vector.
- **mu0**: hypothesized mean $\mu_0$.
- **var0**: hypothesized variance $\sigma^2_0$.

**Value**

a (list) object of S3 class htest containing:

- **statistic**: a test statistic.
- **p.value**: $p$-value under $H_0$.
- **alternative**: alternative hypothesis.
- **method**: name of the test.
- **data.name**: name(s) of provided sample data.
**Examples**

```r
## CRAN-purpose small example
mvar1.LRT(rnorm(10))
```

```r
## Not run:
## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
  x = rnorm(100) # sample x from N(0,1)
  counter[i] = ifelse(mvar1.LRT(x)$p.value < 0.05, 1, 0)
}

## print the result
cat(paste("\nExample for 'mvar1.LRT'\n","\n","* number of rejections : ", sum(counter),"\n","* total number of trials : ", niter,"\n","* empirical Type 1 error : ", round(sum(counter/niter),5),"\n",sep=""))
```

```r
## End(Not run)
```

---

**mvar2.1930PN**

Two-sample Simultaneous Test of Mean and Variance by Pearson and Neyman (1930)

**Description**

Given two univariate samples $x$ and $y$, it tests

$$H_0 : \mu_x = \mu_y, \sigma_x^2 = \sigma_y^2 \ vs \ H_1 : \ not \ H_0$$

by approximating the null distribution with Beta distribution using the first two moments matching.

**Usage**

```r
mvar2.1930PN(x, y)
```

**Arguments**

- `x` a length-$n$ data vector.
- `y` a length-$m$ data vector.
Value

- **statistic**: a test statistic.
- **p.value**: p-value under $H_0$.
- **alternative**: alternative hypothesis.
- **method**: name of the test.
- **data.name**: name(s) of provided sample data.

Examples

```r
## CRAN-purpose small example
x = rnorm(10)
y = rnorm(10)
mvar2.1930PN(x, y)

## Not run:
## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
  x = rnorm(100) # sample x from N(0,1)
y = rnorm(100) # sample y from N(0,1)
  counter[i] = ifelse(mvar2.1930PN(x,y)$p.value < 0.05, 1, 0)
}
## print the result
cat(paste("\n* Example for "mvar2.1930PN\n","\n* number of rejections : ", sum(counter),"\n",
* total number of trials : ", niter,"\n",
* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))

## End(Not run)
```

mvar2.1976PL

**Two-sample Simultaneous Test of Mean and Variance by Perng and Littell (1976)**

Description

Given two univariate samples $x$ and $y$, it tests

$$H_0 : \mu_x = \mu_y, \sigma_x^2 = \sigma_y^2 \text{ vs } H_1 : \text{not } H_0$$

using Fisher’s method of merging two $p$-values.
Usage

mvar2.1976PL(x, y)

Arguments

x a length-\(n\) data vector.

y a length-\(m\) data vector.

Value

a (list) object of S3 class htest containing:

- **statistic** a test statistic.
- **p.value** \(p\)-value under \(H_0\).
- **alternative** alternative hypothesis.
- **method** name of the test.
- **data.name** name(s) of provided sample data.

References


Examples

```r
## CRAN-purpose small example
x = rnorm(10)
y = rnorm(10)
mvar2.1976PL(x, y)

## Not run:
## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
x = rnorm(100) # sample x from N(0,1)
y = rnorm(100) # sample y from N(0,1)

counter[i] = ifelse(mvar2.1976PL(x,y)$p.value < 0.05, 1, 0)
}

## print the result
cat(paste("\n\nExample for 'mvar2.1976PL'\n","*
","* number of rejections : ", sum(counter),"\n",
","* total number of trials : ", niter,"\n",
","* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))

## End(Not run)
```
Description

Given two univariate samples $x$ and $y$, it tests

$$H_0: \mu_x = \mu_y, \sigma^2_x = \sigma^2_y \ vs \ H_1: \ not \ H_0$$

using Muirhead’s approximation for small-sample problem.

Usage

```
mvar2.1982Muirhead(x, y)
```

Arguments

- **x**  
a length-$n$ data vector.
- **y**  
a length-$m$ data vector.

Value

A (list) object of S3 class `htest` containing:

- `statistic` a test statistic.
- `p.value` $p$-value under $H_0$.
- `alternative` alternative hypothesis.
- `method` name of the test.
- `data.name` name(s) of provided sample data.

References


Examples

```r
## CRAN-purpose small example
x = rnorm(10)
y = rnorm(10)
mvar2.1982Muirhead(x, y)
```

## Not run:
```
## empirical Type I error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
```
x = rnorm(100) # sample x from N(0,1)
y = rnorm(100) # sample y from N(0,1)

counter[i] = ifelse(mvar2.1982Muirhead(x,y)$p.value < 0.05, 1, 0)
}

## print the result

cat(paste("nExample for 'mvar2.1982Muirhead'
n","n","number of rejections : ", sum(counter),"n","n","total number of trials : ", niter,"n","n","empirical Type I error : ",round(sum(counter/niter),5),"n",sep=""))

## End(Not run)

mvar2.2012ZXC  Two-sample Simultaneous Test of Mean and Variance by Zhang, Xu, and Chen (2012)

Description

Given two univariate samples x and y, it tests

\[ H_0 : \mu_x = \mu_y, \sigma_x^2 = \sigma_y^2 \quad \text{vs} \quad H_1 : \text{not } H_0 \]

using exact null distribution for likelihood ratio statistic.

Usage

mvar2.2012ZXC(x, y)

Arguments

x a length-n data vector.
y a length-m data vector.

Value

a (list) object of S3 class htest containing:

- **statistic** a test statistic.
- **p.value** p-value under \( H_0 \).
- **alternative** alternative hypothesis.
- **method** name of the test.
- **data.name** name(s) of provided sample data.
References


Examples

```r
## CRAN-purpose small example
x = rnorm(10)
y = rnorm(10)
mvar2.2012ZXC(x, y)

## Not run:
## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
x = rnorm(100) # sample x from N(0,1)
y = rnorm(100) # sample y from N(0,1)
counter[i] = ifelse(mvar2.2012ZXC(x,y)$p.value < 0.05, 1, 0)
print(paste("* mvar2.2012ZXC : iteration ",i,"/",niter," complete.",sep=""))
}
## print the result
cat(paste("\n\n* number of rejections : ", sum(counter),"\n",
* total number of trials : ", niter,"\n",
* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
## End(Not run)
```

mvar2.LRT

Two-sample Simultaneous Likelihood Ratio Test of Mean and Variance

Description

Given two univariate samples \( x \) and \( y \), it tests

\[
H_0 : \mu_x = \mu_y, \sigma^2_x = \sigma^2_y \quad \text{vs} \quad H_1 : \text{not } H_0
\]

using classical likelihood ratio test.

Usage

mvar2.LRT(x, y)

Arguments

- `x`: a length-\( n \) data vector.
- `y`: a length-\( m \) data vector.
Value

a (list) object of S3 class htest containing:

statistic a test statistic.
p.value p-value under $H_0$.
alternative alternative hypothesis.
method name of the test.
data.name name(s) of provided sample data.

Examples

```r
## CRAN-purpose small example
x = rnorm(10)
y = rnorm(10)
mvar2.LRT(x, y)
```

```r
## Not run:
## empirical Type 1 error
niter = 1000
counter = rep(0, niter) # record p-values
for (i in 1:niter){
x = rnorm(100) # sample x from N(0,1)
y = rnorm(100) # sample y from N(0,1)
counter[i] = ifelse(mvar2.LRT(x,y)$p.value < 0.05, 1, 0)
}
## print the result
cat(paste("\n* Example for 'mvar2.LRT' \n","* number of rejections : ", sum(counter),"\n","* total number of trials : ", niter,"\n","* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
```

```r
## End(Not run)
```

norm.1965SW

Univariate Test of Normality by Shapiro and Wilk (1965)

Description

Given an univariate sample $x$, it tests

$$H_0 : x \text{ is from normal distribution } vs \ H_1 : \text{ not } H_0$$

Usage

norm.1965SW(x)

Arguments

x  a length-n data vector.

Value

a (list) object of S3 class htest containing:

- statistic  a test statistic.
- p.value  p-value under \( H_0 \).
- alternative  alternative hypothesis.
- method  name of the test.
- data.name  name(s) of provided sample data.

References


Examples

```r
## generate samples from several distributions
x = stats::runif(28)  # uniform
y = stats::rgamma(28, shape=2)  # gamma
z = stats::rlnorm(28)  # log-normal

## test above samples
test.x = norm.1965SW(x)  # uniform
test.y = norm.1965SW(y)  # gamma
test.z = norm.1965SW(z)  # log-normal
```

Description

Given an univariate sample \( x \), it tests

\[
H_0 : x \text{ is from normal distribution} \quad vs \quad H_1 : \text{ not } H_0
\]

using a test procedure by Shapiro and Francia (1972), which is an approximation to Shapiro and Wilk (1965).
**Usage**

\texttt{norm.1972SF(x)}

**Arguments**

\texttt{x}  
a length-\(n\) data vector.

**Value**

A (list) object of S3 class \texttt{htest} containing:

- \texttt{statistic}  
a test statistic.
- \texttt{p.value}  
p-value under \(H_0\).
- \texttt{alternative}  
alternative hypothesis.
- \texttt{method}  
name of the test.
- \texttt{data.name}  
name(s) of provided sample data.

**References**


**Examples**

```r
## CRAN-purpose small example
x = rnorm(10)
norm.1972SF(x) # run the test

## generate samples from several distributions
x = stats::runif(496)  # uniform
y = stats::rgamma(496, shape=2)  # gamma
z = stats::rlnorm(496)  # log-normal

## test above samples
test.x = norm.1972SF(x) # uniform
test.y = norm.1972SF(y) # gamma
test.z = norm.1972SF(z) # log-normal
```
Univariate Test of Normality by Jarque and Bera (1980)

Description

Given an univariate sample $x$, it tests

$$H_0 : x \text{ is from normal distribution} \quad vs \quad H_1 : \text{ not } H_0$$

using a test procedure by Jarque and Bera (1980).

Usage

```r
norm.1980JB(x, method = c("asymptotic", "MC"), nreps = 2000)
```

Arguments

- `x` a length-$n$ data vector.
- `method` method to compute $p$-value. Using initials is possible, "a" for asymptotic for example. Case insensitive.
- `nreps` the number of Monte Carlo simulations to be run when `method="MC"`.

Value

a (list) object of S3 class `htest` containing:

- `statistic` a test statistic.
- `p.value` $p$-value under $H_0$.
- `alternative` alternative hypothesis.
- `method` name of the test.
- `data.name` name(s) of provided sample data.

References


Examples

```r
## generate samples from uniform distribution
x = runif(28)

## test with both methods of attaining p-values
test1 = norm.1980JB(x, method="a") # Asymptotics
test2 = norm.1980JB(x, method="m") # Monte Carlo
```
Adjusted Jarque-Bera Test of Univariate Normality by Urzua (1996)

Description

Given an univariate sample \( x \), it tests

\[
H_0 : \text{\( x \) is from normal distribution} \quad \text{vs} \quad H_1 : \text{not} \ H_0
\]

using a test procedure by Urzua (1996), which is a modification of Jarque-Bera test.

Usage

\[
\text{norm.1996AJB}(x, \text{method} = \text{c("asymptotic", "MC")}, \text{nreps} = 2000)
\]

Arguments

\( x \) a length-\( n \) data vector.
\( \text{method} \) method to compute \( p \)-value. Using initials is possible, "a" for asymptotic for example.
\( \text{nreps} \) the number of Monte Carlo simulations to be run when \( \text{method} = \text{"MC"} \).

Value

a (list) object of S3 class \( \text{htest} \) containing:

\( \text{statistic} \) a test statistic.
\( \text{p.value} \) \( p \)-value under \( H_0 \).
\( \text{alternative} \) alternative hypothesis.
\( \text{method} \) name of the test.
\( \text{data.name} \) name(s) of provided sample data.

References


Examples

\[
\text{## generate samples from uniform distribution}
\text{x = runif(28)}
\]
\[
\text{## test with both methods of attaining p-values}
\text{test1 = norm.1996AJB(x, method="a") # Asymptotics}
\text{test2 = norm.1996AJB(x, method="m") # Monte Carlo}
\]
Robust Jarque-Bera Test of Univariate Normality by Gel and Gastwirth (2008)

Description

Given an univariate sample \( x \), it tests

\[
H_0 : x \text{ is from normal distribution} \quad \text{vs} \quad H_1 : \text{not } H_0
\]

using a test procedure by Gel and Gastwirth (2008), which is a robustified version Jarque-Bera test.

Usage

\[
\text{norm.2008RJB}(x, \ C1 = 6, \ C2 = 24, \ \text{method} = \text{c("asymptotic", "MC"), \ nreps = 2000})
\]

Arguments

- \( x \) a length-\( n \) data vector.
- \( C1 \) a control constant. Authors proposed \( C1 = 6 \) for nominal level of \( \alpha = 0.05 \).
- \( C2 \) a control constant. Authors proposed \( C2 = 24 \) for nominal level of \( \alpha = 0.05 \).
- \( \text{method} \) method to compute \( p \)-value. Using initials is possible, "a" for asymptotic for example.
- \( \text{nreps} \) the number of Monte Carlo simulations to be run when \text{method}="MC".

Value

a (list) object of S3 class \text{htest} containing:

- \text{statistic} a test statistic.
- \text{p.value} \( p \)-value under \( H_0 \).
- \text{alternative} alternative hypothesis.
- \text{method} name of the test.
- \text{data.name} name(s) of provided sample data.

References


Examples

```r
## generate samples from uniform distribution
x = runif(28)
## test with both methods of attaining p-values
test1 = norm.2008RJB(x, method="a") # Asymptotics
test2 = norm.2008RJB(x, method="m") # Monte Carlo
```
One-sample Simultaneous Test of Mean and Covariance by Liu et al. (2017)

Description

Given a multivariate sample $X$, hypothesized mean $\mu_0$ and covariance $\Sigma_0$, it tests

$$H_0 : \mu_x = \mu_0 \text{ and } \Sigma_x = \Sigma_0 \quad \text{vs} \quad H_1 : \text{not } H_0$$

using the procedure by Liu et al. (2017).

Usage

```r
sim1.2017Liu(X, mu0 = rep(0, ncol(X)), Sigma0 = diag(ncol(X)))
```

Arguments

- `X` an $(n \times p)$ data matrix where each row is an observation.
- `mu0` a length-$p$ mean vector of interest.
- `Sigma0` a $(p \times p)$ given covariance matrix.

Value

a (list) object of S3 class `htest` containing:

- `statistic` a test statistic.
- `p.value` $p$-value under $H_0$.
- `alternative` alternative hypothesis.
- `method` name of the test.
- `data.name` name(s) of provided sample data.

References


Examples

```r
## CRAN-purpose small example
smallX = matrix(rnorm(10*3),ncol=3)
sim1.2017Liu(smallX) # run the test
```

```r
## Not run:
## empirical Type 1 error
niter = 1000
```
counter = rep(0,niter)  # record p-values
for (i in 1:niter){
  X = matrix(rnorm(50*10), ncol=10)
  counter[i] = ifelse(sim1.2017Liu(X)$p.value < 0.05, 1, 0)
}

## print the result
cat(paste("\n* Example for \"sim1.2017Liu\"*, \n\n* number of rejections : ", sum(counter), "\n",
* total number of trials : ", niter,"\n",
* empirical Type 1 error : ",round(sum(counter/niter),5), "\n",sep="\n"))

## End(Not run)

---

**sim1.LRT**

*One-sample Simultaneous Likelihood Ratio Test of Mean and Covariance*

**Description**

Given a multivariate sample $X$, hypothesized mean $\mu_0$ and covariance $\Sigma_0$, it tests

$$H_0 : \mu_x = \mu_0 \text{ and } \Sigma_x = \Sigma_0 \quad vs \quad H_1 : \text{ not } H_0$$

using the standard likelihood-ratio test procedure.

**Usage**

```r
sim1.LRT(X, mu0 = rep(0, ncol(X)), Sigma0 = diag(ncol(X)))
```

**Arguments**

- `X` an $(n \times p)$ data matrix where each row is an observation.
- `mu0` a length-$p$ mean vector of interest.
- `Sigma0` a $(p \times p)$ given covariance matrix.

**Value**

a (list) object of S3 class `htest` containing:

- `statistic` a test statistic.
- `p.value` $p$-value under $H_0$.
- `alternative` alternative hypothesis.
- `method` name of the test.
- `data.name` name(s) of provided sample data.
## CRAN-purpose small example

```r
smallX = matrix(rnorm(10*3),ncol=3)
sim1.LRT(smallX) # run the test
```

## Not run:

```r
## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
  X = matrix(rnorm(100*10), ncol=10)
  counter[i] = ifelse(sim1.LRT(X)$p.value < 0.05, 1, 0)
  print(paste("* iteration ",i,"/1000 complete..."))
}
```

## print the result

```r
cat(paste("\n* Example for \'sim1.LRT\'\n","\n","* number of rejections : ", sum(counter),"\n",
","* total number of trials : ", niter,"\n",
","* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
```

## End(Not run)

---

### Two-sample Simultaneous Test of Means and Covariances by Hyodo and Nishiyama (2018)

#### Description

Given a multivariate sample $X$, hypothesized mean $\mu_0$ and covariance $\Sigma_0$, it tests

$$ H_0 : \mu_x = \mu_y \text{ and } \Sigma_x = \Sigma_y \quad \text{vs} \quad H_1 : \text{not } H_0 $$

using the procedure by Hyodo and Nishiyama (2018) in a similar fashion to that of Liu et al. (2017) for one-sample test.

#### Usage

```r
sim2.2018HN(X, Y)
```

#### Arguments

- **X** an $(n_x \times p)$ data matrix of 1st sample.
- **Y** an $(n_y \times p)$ data matrix of 2nd sample.
Value

a (list) object of S3 class htest containing:

- **statistic** a test statistic.
- **p.value** p-value under $H_0$.
- **alternative** alternative hypothesis.
- **method** name of the test.
- **data.name** name(s) of provided sample data.

References


Examples

```r
## CRAN-purpose small example
smallX = matrix(rnorm(10*3),ncol=3)
smallY = matrix(rnorm(10*3),ncol=3)
sim2.2018HN(smallX, smallY) # run the test

## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
  X = matrix(rnorm(121*10), ncol=10)
  Y = matrix(rnorm(169*10), ncol=10)
  counter[i] = ifelse(sim2.2018HN(X,Y)$p.value < 0.05, 1, 0)
}

## print the result
cat(paste("\n* Example for '\"Var sim2.2018HN\"Var
","*
","* number of rejections : ", sum(counter),"\n",
","* total number of trials : ", niter,"\n",
","* empirical Type 1 error : "",round(sum(counter/niter),5),"\n",sep=""))
```

---

simplex.uniform

*Probability Simplex : Tests of Uniformity*

Description

Given a data $X \in \mathbb{R}^n \times p$ such that its rows are vectors in a probability simplex, i.e., $x \in \Delta_{p-1} = \{z \in \mathbb{R}^p \mid z_j > 0, \sum_{i=1}^p z_i = 1\}$, test whether the data is uniformly distributed.
Usage

simplex.uniform(X, method)

Arguments

X an \((n \times p)\) data matrix where each row is an observation.

method \((case-insensitive)\) name of the method to be used, including

- **LRT** likelihood-ratio test with the Dirichlet distribution.
- **LRTsym** likelihood-ratio test using the symmetric Dirichlet distribution (default).

Value

a (list) object of S3 class `htest` containing:

- **statistic** a test statistic.
- **p.value** \(p\)-value under \(H_0\).
- **alternative** alternative hypothesis.
- **method** name of the test.
- **data.name** name(s) of provided sample data.

Examples

```r
## pseudo-uniform data generation
N = 100
P = 4
X = matrix(stats::rnorm(N*P), ncol=P)
for (n in 1:N){
  x = X[n,]
  x = abs(x/sqrt(sum(x^2)))
  X[n,] = x^2
}
## run the tests
simplex.uniform(X, "LRT")
simplex.uniform(X, "lrtsym")
```
Description

Given a multivariate sample $X$, it tests

$$H_0 : \Sigma_x = \text{uniform on } \bigotimes_{i=1}^{p} [a_i, b_i] \quad \text{vs} \quad H_1 : \text{not } H_0$$

using the procedure by Yang and Modarres (2017). Originally, it tests the goodness of fit on the unit hypercube $[0,1]^p$ and modified for arbitrary rectangular domain.

Usage

```
unif.2017YMi(
  x,
  type = c("Q1", "Q2", "Q3"),
  lower = rep(0, ncol(x)),
  upper = rep(1, ncol(x))
)
```

Arguments

- `x` an $(n \times p)$ data matrix where each row is an observation.
- `type` type of statistic to be used, one of "Q1","Q2", and "Q3".
- `lower` length-$p$ vector of lower bounds of the test domain.
- `upper` length-$p$ vector of upper bounds of the test domain.

Value

a (list) object of S3 class `htest` containing:

- `statistic` a test statistic.
- `p.value` $p$-value under $H_0$.
- `alternative` alternative hypothesis.
- `method` name of the test.
- `data.name` name(s) of provided sample data.

References

Examples

```r
## CRAN-purpose small example
smallX = matrix(rnorm(10*3),ncol=3)
unif.2017YMq(smallX) # run the test

## empirical Type 1 error
## compare performances of three methods
niter = 1234
rec1 = rep(0,niter) # for Q1
rec2 = rep(0,niter) # Q2
rec3 = rep(0,niter) # Q3
for (i in 1:niter){
  X = matrix(runif(50*10), ncol=50) # (n,p) = (10,50)
  rec1[i] = ifelse(unif.2017YMq(X, type="Q1")$p.value < 0.05, 1, 0)
  rec2[i] = ifelse(unif.2017YMq(X, type="Q2")$p.value < 0.05, 1, 0)
  rec3[i] = ifelse(unif.2017YMq(X, type="Q3")$p.value < 0.05, 1, 0)
}
## print the result
cat(paste("\n* Example for 'unif.2017YMq', \n"*, "round(sum(rec1/niter),5),"\n",
"* Type 1 error with Q1 : ", round(sum(rec1/niter),5),"\n",
"* Q2 : ", round(sum(rec2/niter),5),"\n",
"* Q3 : ", round(sum(rec3/niter),5),"\n",sep=""))
```

**Description**

Given a multivariate sample $X$, it tests

$$H_0 : \Sigma X = \text{uniform on } \otimes_{i=1}^{P} [a_i, b_i] \quad \text{vs} \quad H_1 : \text{not } H_0$$

using the procedure by Yang and Modarres (2017). Originally, it tests the goodness of fit on the unit hypercube $[0, 1]^P$ and modified for arbitrary rectangular domain. Since this method depends on quantile information, every observation should strictly reside within the boundary so that it becomes valid after transformation.

**Usage**

```r
unif.2017YMq(X, lower = rep(0, ncol(X)), upper = rep(1, ncol(X)))
```

**Arguments**

- **X**
  - an $(n \times p)$ data matrix where each row is an observation.
- **lower**
  - length-$p$ vector of lower bounds of the test domain.
- **upper**
  - length-$p$ vector of upper bounds of the test domain.
Value

a (list) object of S3 class htest containing:

- **statistic** a test statistic.
- **p.value** p-value under $H_0$.
- **alternative** alternative hypothesis.
- **method** name of the test.
- **data.name** name(s) of provided sample data.

References


Examples

```r
## CRAN-purpose small example
smallX = matrix(runif(10*3),ncol=3)
unif.2017YMq(smallX) # run the test

## empirical Type 1 error
niter = 1234
counter = rep(0,niter) # record p-values
for (i in 1:niter){
  X = matrix(runif(50*5), ncol=25)
  counter[i] = ifelse(unif.2017YMq(X)$p.value < 0.05, 1, 0)
}
## print the result
cat(paste("\n\n* Example for 'unif.2017YMq'\n"="\n","* number of rejections : ", sum(counter),"\n",
"* total number of trials : ", niter,"\n",
"* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
```

---

**usek1d**

*Apply k-sample tests for two univariate samples*

**Description**

Any k-sample method implies that it can be used for a special case of $k = 2$. **usek1d** lets any k-sample tests provided in this package be used with two univariate samples $x$ and $y$.

**Usage**

```r
usek1d(x, y, test.name, ...)
```
useknd

Apply k-sample tests for two multivariate samples

Arguments

- **x** a length-\( n \) data vector.
- **y** a length-\( m \) data vector.
- **test.name** character string for the name of k-sample test to be used.
- **...** extra arguments passed onto the function **test.name**.

Value

a (list) object of S3 class **htest** containing:

- **statistic** a test statistic.
- **p.value** \( p \)-value under \( H_0 \).
- **alternative** alternative hypothesis.
- **method** name of the test.
- **data.name** name(s) of provided sample data.

Examples

```r
### compare two-means via anova and t-test
### since they coincide when k=2
x = rnorm(50)
y = rnorm(50)

### run anova and t-test
test1 = usekld(x, y, "meank.anova")
test2 = mean2.ttest(x,y)

## print the result
cat(paste("\n* Comparison of ANOVA and t-test \n","* ",
"* p-value from ANOVA : ", round(test1$p.value,5),"\n",
"* t-test : " , round(test2$p.value,5),"\n",sep=""))
```

Description

Any \( k \)-sample method implies that it can be used for a special case of \( k = 2 \). **useknd** lets any \( k \)-sample tests provided in this package be used with two multivariate samples \( X \) and \( Y \).

Usage

**useknd(X, Y, test.name, ...)**
Arguments

X an \((n_x \times p)\) data matrix of 1st sample.
Y an \((n_y \times p)\) data matrix of 2nd sample.
test.name character string for the name of k-sample test to be used.
... extra arguments passed onto the function test.name.

Value

a (list) object of S3 class htest containing:

statistic a test statistic.
p.value \(p\)-value under \(H_0\).
alternative alternative hypothesis.
method name of the test.
data.name name(s) of provided sample data.

Examples

## use 'covk.2007Schott' for two-sample covariance testing
## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
  X = matrix(rnorm(50*5), ncol=10)
  Y = matrix(rnorm(50*5), ncol=10)
  counter[i] = ifelse(useknd(X,Y,"covk.2007Schott")$p.value < 0.05, 1, 0)
}
## print the result
cat(paste("\n* Example for 'covk.2007Schott'\n","* number of rejections : ", sum(counter),"\n",
  "* total number of trials : ", niter,"\n",
  "* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))

---

\textbf{var1.chisq} \hspace{1cm} \textit{One-Sample Chi-Square Test for Variance}

\textbf{Description}

Given an univariate sample \(x\), it tests

\[ H_0 : \sigma^2_x \{=,\geq,\leq\} \sigma^2_0 \hspace{0.5cm} vs \hspace{0.5cm} H_1 : \sigma^2_x \{\neq, <, >\} \sigma^2_0 \]
Usage

```r
var1.chisq(x, var0 = 1, alternative = c("two.sided", "less", "greater"))
```

Arguments

- **x**: a length-$n$ data vector.
- **var0**: hypothesized variance $\sigma_0^2$.
- **alternative**: specifying the alternative hypothesis.

Value

A (list) object of S3 class *htest* containing:

- **statistic**: a test statistic.
- **p.value**: $p$-value under $H_0$.
- **alternative**: alternative hypothesis.
- **method**: name of the test.
- **data.name**: name(s) of provided sample data.

References


Examples

```r
## CRAN-purpose small example
x = rnorm(10)
var1.chisq(x, alternative="g") ## Ha : var(x) >= 1
var1.chisq(x, alternative="l") ## Ha : var(x) <= 1
var1.chisq(x, alternative="t") ## Ha : var(x) /= 1

## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
  x = rnorm(50) # sample x from N(0,1)
  counter[i] = ifelse(var1.chisq(x, var0=1)$p.value < 0.05, 1, 0)
}

## print the result
cat(paste("\n* Example for 'var1.chisq'\n","\n* number of rejections : ", sum(counter),"\n","* total number of trials : ", niter,"\n","* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
```
Description

Given two univariate samples \(x\) and \(y\), it tests

\[
H_0 : \sigma^2_x \{=, \geq, \leq\} \sigma^2_y \space vs \space H_1 : \sigma^2_x \{\neq, <, >\} \sigma^2_y
\]

Usage

\[
\text{var2.F}(x, y, \text{alternative} = \text{c("two.sided", "less", "greater")})
\]

Arguments

- \(x\) a length-\(n\) data vector.
- \(y\) a length-\(m\) data vector.
- \(\text{alternative}\) specifying the alternative hypothesis.

Value

a (list) object of \(S3\) class \(\text{htest}\) containing:

- \text{statistic} a test statistic.
- \text{p.value} \(p\)-value under \(H_0\).
- \text{alternative} alternative hypothesis.
- \text{method} name of the test.
- \text{data.name} name(s) of provided sample data.

References


Examples

\[
\begin{align*}
\text{## CRAN-purpose small example} \\
\text{x} &= \text{rnorm(10)} \\
\text{y} &= \text{rnorm(10)} \\
\text{var2.F}(x, y, \text{alternative}="g") \ & \text{## Ha : var(x) >= var(y)} \\
\text{var2.F}(x, y, \text{alternative}="l") \ & \text{## Ha : var(x) <= var(y)} \\
\text{var2.F}(x, y, \text{alternative}="t") \ & \text{## Ha : var(x) /= var(y)} \\
\end{align*}
\]

\[
\begin{align*}
\text{## empirical Type 1 error}
\end{align*}
\]
niter = 1000

counter = rep(0,niter) # record p-values

for (i in 1:niter){
  x = rnorm(57) # sample x from N(0,1)
  y = rnorm(89) # sample y from N(0,1)

  counter[i] = ifelse(var2.F(x,y)$p.value < 0.05, 1, 0)
}

## print the result

## Example for Var

vark.1937Bartlett

Bartlett's Test for Homogeneity of Variance

Description

Given univariate samples \(X_1, \ldots, X_k\), it tests

\[
H_0 : \sigma^2_1 = \cdots = \sigma^2_k \quad \text{vs} \quad H_1 : \text{at least one equality does not hold}
\]

using the procedure by Bartlett (1937).

Usage

vark.1937Bartlett(dlist)

Arguments

dlist a list of length \(k\) where each element is a sample vector.

Value

a (list) object of S3 class htest containing:

- **statistic** a test statistic.
- **p.value** \(p\)-value under \(H_0\).
- **alternative** alternative hypothesis.
- **method** name of the test.
- **data.name** name(s) of provided sample data.

References

Examples

```r
## CRAN-purpose small example
small1d = list()
for (i in 1:5){  # k=5 sample
  small1d[[i]] = rnorm(20)
}
vark.1937Bartlett(small1d)  # run the test

## test when k=5 (samples)
## empirical Type 1 error
niter = 1000
counter = rep(0,niter)  # record p-values
for (i in 1:niter){
  mylist = list()
  for (j in 1:5){
    mylist[[j]] = rnorm(50)
  }
  counter[i] = ifelse(vark.1937Bartlett(mylist)$p.value < 0.05, 1, 0)
}

## print the result
cat(paste("\n\n* Example for ‘vark.1937Bartlett’\n","\n","* number of rejections : ", sum(counter),"\n","* total number of trials : ", niter,"\n","* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
```

vark.1960Levene

Levene’s Test for Homogeneity of Variance

Description

Given univariate samples $X_1, \ldots, X_k$, it tests

$$H_0 : \sigma_1^2 = \cdots = \sigma_k^2 \ vs \ H_1 : \text{at least one equality does not hold}$$

using the procedure by Levene (1960).

Usage

`vark.1960Levene(dlist)`

Arguments

dlist a list of length $k$ where each element is a sample vector.
Value

a (list) object of S3 class htest containing:

- **statistic** a test statistic.
- **p.value** $p$-value under $H_0$.
- **alternative** alternative hypothesis.
- **method** name of the test.
- **data.name** name(s) of provided sample data.

References


Examples

```r
## CRAN-purpose small example
small1d = list()
for (i in 1:5){ # k=5 sample
  small1d[[i]] = rnorm(20)
}
vark.1960Levene(small1d) # run the test

## test when k=5 (samples)
## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
  mylist = list()
  for (j in 1:5){
    mylist[[j]] = rnorm(50)
  }
  counter[i] = ifelse(vark.1960Levene(mylist)$p.value < 0.05, 1, 0)
}

## print the result
cat(paste("\n* Example for 'vark.1960Levene' \n","\n","\n* number of rejections : ", sum(counter), "\n","\n* total number of trials : ", niter, "\n","\n* empirical Type 1 error : ",round(sum(counter/niter),5), "\n",sep=""))
```
Brown-Forsythe Test for Homogeneity of Variance

Description

Given univariate samples \(X_1, \ldots, X_k\), it tests

\[ H_0 : \sigma_1^2 = \cdots = \sigma_k^2 \quad \text{vs} \quad H_1 : \text{at least one equality does not hold} \]


Usage

vark.1974BF(dlist)

Arguments

dlist a list of length \(k\) where each element is a sample vector.

Value

a (list) object of S3 class htest containing:

- statistic a test statistic.
- p.value \(p\)-value under \(H_0\).
- alternative alternative hypothesis.
- method name of the test.
- data.name name(s) of provided sample data.

References


Examples

```r
## CRAN-purpose small example
small1d = list()
for (i in 1:5){ # k=5 sample
  small1d[[i]] = rnorm(20)
}
vark.1974BF(small1d) # run the test

## test when k=5 (samples)
## empirical Type 1 error
niter = 1000
counter = rep(0,niter) # record p-values
```
for (i in 1:niter){
    mylist = list()
    for (j in 1:5){
        mylist[[j]] = rnorm(50)
    }

    counter[i] = ifelse(vark.1974BF(mylist)$p.value < 0.05, 1, 0)
}

## print the result
cat(paste("\n* Example for 'vark.1974BF'\n","\n","* number of rejections : ", sum(counter),"\n","* total number of trials : ", niter,"\n","* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
Index

* covariance
  cov1.2012Fisher, 3
  cov1.2015WL, 4
  cov2.2012LC, 5
  cov2.2013CLX, 7
  cov2.2015WL, 8
  cokv.2011Schott, 9
  cokv.2007Schott, 11

* eqdist
  eqdist.2014BG, 12

* gof_normal
  norm.1965SW, 51
  norm.1972SF, 52
  norm.1980JB, 54
  norm.1996AJB, 55
  norm.2008RJB, 56

* gof_uniform
  unif.2017YMi, 62
  unif.2017YMq, 63

* mean_multivariate
  mean1.1931Hotelling, 14
  mean1.1958Dempster, 15
  mean1.1996BS, 16
  mean1.2008SD, 17
  mean2.1931Hotelling, 20
  mean2.1958Dempster, 21
  mean2.1965Yao, 22
  mean2.1980Johansen, 24
  mean2.1986NVM, 25
  mean2.1996BS, 26
  mean2.2004KY, 28
  mean2.2008SD, 29
  mean2.2011JW, 30
  mean2.2014CLX, 32
  mean2.2014Thulin, 33
  mean2.mxPBF, 35
  meank.2007Schott, 37
  meank.20092Z, 39
  meank.2019CPH, 40

* mean_univariate
  mean1.ttest, 18
  mean2.ttest, 36
  meank.anova, 42

* mvar
  mvar1.1998AS, 43
  mvar1.LRT, 44
  mvar2.1930PN, 45
  mvar2.1976PL, 46
  mvar2.1982Muirhead, 48
  mvar2.2012ZXC, 49
  mvar2.LRT, 50

* simplex
  simplex.uniform, 60

* simtest
  sim1.2017Liu, 57
  sim1.LRT, 58
  sim2.2018HN, 59

* utility
  usek1d, 64
  useknd, 65

* variance
  var1.chisq, 66
  var2.F, 68
  vark.1937Bartlett, 69
  vark.1960Levene, 70
  vark.1974BF, 72

  cov1.2012Fisher, 3
  cov1.2015WL, 4
  cov2.2012LC, 5
  cov2.2013CLX, 7
  cov2.2015WL, 8
  cokv.2001Schott, 9
  cokv.2007Schott, 11

  eqdist.2014BG, 12

  mean1.1931Hotelling, 14
  mean1.1958Dempster, 15
INDEX

mean1.1996BS, 16
mean1.2008SD, 17
mean1.ttest, 18
mean2.1931Hotelling, 20
mean2.1958Dempster, 21
mean2.1965Yao, 22
mean2.1980Johansen, 24
mean2.1986NVM, 25
mean2.1996BS, 26
mean2.2004KY, 28
mean2.2008SD, 29
mean2.2011LJW, 30
mean2.2014CLX, 32
mean2.2014Thulin, 33
mean2.mxPBF, 35
mean2.ttest, 36
meank.2007Schott, 37
meank.2009ZX, 39
meank.2019CPH, 40
meank.anova, 42
mvar1.1998AS, 43
mvar1.LRT, 44
mvar2.1930PN, 45
mvar2.1976PL, 46
mvar2.1982Muirhead, 48
mvar2.2012ZXC, 49
mvar2.LRT, 50

norm.1965SW, 51
norm.1972SF, 52
norm.1980JB, 54
norm.1996AJB, 55
norm.2008RJB, 56

sim1.2017Liu, 57
sim1.LRT, 58
sim2.2018HN, 59
simplex.uniform, 60

unif.2017YM, 62
unif.2017YMq, 63
usek1d, 64
useknd, 65

var1.chisq, 66
var2.F, 68
vark.1937Bartlett, 69
vark.1960Levene, 70
vark.1974BF, 72