Package ‘SHT’
July 4, 2021

Type Package
Title Statistical Hypothesis Testing Toolbox
Version 0.1.5
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://kisungyou.com/SHT/
BugReports https://github.com/kisungyou/SHT/issues
Imports Rcpp, Rdpack, stats, utils, fastclime, pracma
LinkingTo Rcpp, RcppArmadillo
RoxygenNote 7.1.1
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 2021-07-04 21:00:09 UTC

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

cov2.mxPBF .................................................. 9

covk.2001Schott ........................................... 11

covk.2007Schott ............................................ 12

eqdist.2014BG ............................................... 13

mean1.1931Hotelling ...................................... 15

mean1.1958Dempster ....................................... 16

mean1.1996BS ............................................... 17

mean1.2008SD ............................................... 18

mean1.ttest ................................................ 19

mean2.1931Hotelling ...................................... 21

mean2.1958Dempster ....................................... 22

mean2.1965Yao ............................................. 23

mean2.1980Johansen ...................................... 25

mean2.1986NVM ............................................ 26

mean2.1996BS ............................................... 27

mean2.2004KY ............................................. 29

mean2.2008SD ............................................... 30

mean2.2011LJW ............................................ 31

mean2.2014CLX ............................................ 33

mean2.2014Thulin ......................................... 34

mean2.mxPBF ............................................... 36

mean2.ttest ................................................ 37

meank.2007Schott ......................................... 38

meank.2009ZX ............................................. 40

meank.2019CPH ............................................ 41

meank.anova ............................................... 43

mvar1.1998AS ............................................. 44

mvar1.LRT ................................................ 45

mvar2.1930PN ............................................. 46

mvar2.1976PL ............................................. 47

mvar2.1982Muirhead ...................................... 49

mvar2.2012ZXC ............................................ 50

mvar2.LRT ................................................ 51

norm.1965SW ............................................. 52

norm.1972SF ............................................. 53

norm.1980JB ............................................. 55

norm.1996AJB ............................................. 56

norm.2008RJB ............................................. 57

sim1.2017Liu ............................................. 58

sim2.2018HN ............................................. 59

unif.2017YMq ............................................ 62

usek1d .................................................. 64

useknd .................................................. 64

var1.chisq ............................................... 65

var2.F .................................................. 67

vark.1937Bartlett ....................................... 68

vark.1960Levene ......................................... 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

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

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.
- **Sigma0**: a $(p \times p)$ given covariance matrix.
- **type**: 1 or 2 for corresponding statistic from the paper.

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

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
smallX = matrix(rnorm(10*3), ncol=3)
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","\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

\[
H_0 : \Sigma_x = \Sigma_0 \quad \text{vs} \quad H_1 : \Sigma_x \neq \Sigma_0
\]

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

```r
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

```r
## CRAN-purpose small example
smallX = matrix(rnorm(10*3),ncol=3)
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(50*10), ncol=50) # (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("* 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=""))
```
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 Li and Chen (2012).

Usage

cov2.2012LC(X, Y, unbiased = FALSE)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>an $(n_x \times p)$ data matrix of 1st sample.</td>
</tr>
<tr>
<td>Y</td>
<td>an $(n_y \times p)$ data matrix of 2nd sample.</td>
</tr>
<tr>
<td>unbiased</td>
<td>a logical; FALSE to use up to 4th-order U-statistics as proposed in the paper, TRUE for faster run under an assumption that $\mu_h = 0$ (default: FALSE).</td>
</tr>
</tbody>
</table>

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)
cov2.2012LC(smallX, smallY) # run the test
```

```r
## 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,unbiased=TRUE)$p.value < 0.05, 1, 0)
```
cov2.2013CLX

print(paste0("iteration ",i,"/1000 complete."))
}

## print the result
cat(paste("n* Example for \'cov2.2012LC\'\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=""))

## 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

$$H_0 : \Sigma_x = \Sigma_y \hspace{1cm} vs \hspace{1cm} H_1 : \Sigma_x \neq \Sigma_y$$

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

**Usage**

```r
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**

## CRAN-purpose small example

```r
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

```r
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

```r
cat(paste("\nExample for 'cov2.2013CLX'\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="")
```

---

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

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

using the procedure by Wu and Li (2015).

### Usage

```r
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","\n","* number of rejections : ", sum(counter),"\n","* total number of trials : ", niter,"\n","* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
```

---

cov2.mxPBF

Two-sample Covariance Test with Maximum Pairwise Bayes Factor

Description

Not Written Here - No Reference Yet.
Usage

cov2.mxPBF(X, Y, a0 = 2, b0 = 2, 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.
b0 scale parameter for inverse-gamma prior.
gamma non-negative variance scaling parameter.
nthreads number of threads for parallel execution via OpenMP.

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.mat matrix of pairwise Bayes factors in natural log.

Examples

## Not run:
## empirical Type 1 error with BF threshold = 20
niter = 12345
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.mxPBF(X,Y)$statistic > 20, 1, 0)
}
## print the result
cat(paste("\n* Example for 'cov2.mxPBF'\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 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 (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 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


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 I 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)
    }
    counter[i] = ifelse(covk.2001Schott(mylist)$p.value < 0.05, 1, 0)
}

## print the result
cat(paste("\n* Example for "covk.2001Schott"\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)

---

covk.2007Schott  
Test for Homogeneity of Covariances by Schott (2007)

### Description

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

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

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("\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=""))
```

---

eqdist.2014BG

*Test for Equality of Two Distributions by Biswas and Ghosh (2014)*

**Description**

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

$$ H_0 : F_X = F_Y \quad vs \quad H_1 : F_X \neq F_Y $$

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**

```r
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 H0.
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)
eqdist.2014BG(smallX, smallY) # run the test

## 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 1 ERROR COMPARISON 
","* Asymptotics : 
","* Permutation :
","\n","\n","\n",round(sum(pval.a/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

```r
mean1.1931Hotelling(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)
mean1.1931Hotelling(smallX) # run the test

## 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)
```
### 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

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

using the procedure by Dempster (1958, 1960).

#### Usage

```r
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("\nExample for 'mean1.1958Dempster'\n","*
","* number of rejections : ", sum(counter),"\n",
","* total number of trials : ", niter,"\n",
","* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
```

---

**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

$$H_0 : \mu_X = \mu_0 \quad vs \quad H_1 : \mu_X \neq \mu_0$$

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=""))
```

---

### mean1.2008SD

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

#### Description

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

\[
H_0 : \mu_x = \mu_0 \quad \text{vs} \quad H_1 : \mu_x \neq \mu_0
\]

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

## 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
print(paste("* 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

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

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

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

using the procedure by Hotelling (1931).

**Usage**

```r
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**

```r
## 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
```

```r
## generate two samples from standard normal distributions.
X = matrix(rnorm(50*5), ncol=5)
```
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.
**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 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 \texttt{S3} class \texttt{htest} containing:

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

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

## 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)
}
## print the result
cat(paste("\n* Example for 'mean2.1965Yao'\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=""))
```
Two-sample Test for Multivariate Means by Johansen (1980)

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 Johansen (1980) by adapting Welch-James approximation of the degree of freedom for Hotelling's \( T^2 \) test.

Usage

\texttt{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 \( \text{S3} \) class \( \text{htest} \) containing:

- \textbf{statistic} a test statistic.
- \textbf{p.value} \( p \)-value under \( H_0 \).
- \textbf{alternative} alternative hypothesis.
- \textbf{method} name of the test.
- \textbf{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
```

```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=10)
Y = matrix(rnorm(50*5), ncol=10)

counter[i] = ifelse(mean2.1980Johansen(X,Y)$p.value < 0.05, 1, 0)

## print the result
cat(paste("\n* Example for 'mean2.1980Johansen'\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)

mean2.1986NVM 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

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

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:

\textbf{statistic} a test statistic.
\textbf{p.value} \(p\)-value under \(H_0\).
\textbf{alternative} alternative hypothesis.
\textbf{method} name of the test.
\textbf{data.name} name(s) of provided sample data.
mean2.1996BS

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 'mean2.1986NVM'\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")
```

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

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

using the procedure by Bai and Saranadasa (1996).

Usage

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

## 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* Example for 'mean2.1996BS'\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=""))
mean2.2004KY

Two-sample Test for Multivariate Means by Krishnamoorthy and Yu (2004)

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

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

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 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(mean2.2004KY(X,Y)$p.value < 0.05, 1, 0)
}

## print the result
cat(paste("Example for\'mean2.2004KYN\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

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

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","*\n",
"* number of rejections : ", sum(counter),"\n",
"* total number of trials : ", niter,"\n",
"* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
```

---

**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

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

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.

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.

nreps the number of permutation iterations 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
## 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

## 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)
}
## print the result
print(paste("* 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

\[
\text{mean2.2014CLX}( \ X, \ Y, \ \text{precision} = \text{c("sparse", "unknown")}, \ \delta = 2, \ \Omega = \text{NULL}, \ cov\.\text{equal} = \text{TRUE} )
\]

Arguments

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

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

\text{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.

\text{cov.equal} a logical to determine homogeneous covariance assumption.

Value

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

\text{statistic} a test statistic.

\text{p.value} \textit{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*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("* Example for \"mean2.2014CLX\"\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 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.
\(n\text{reps}\) 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
```

```r
## 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)

## run both methods with 100 permutations
mean2.2011LJW(X,Y,nreps=100,method="m") # 2011LJW requires 'm' to be set.
mean2.2014Thulin(X,Y,nreps=100)
```
mean2.mxPBF  

Two-sample Mean Test with Maximum Pairwise Bayes Factor

Description
Not Written Here - No Reference Yet.

Usage
mean2.mxPBF(X, Y, a0 = 2, b0 = 2, 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.
b0  scale parameter for inverse-gamma prior.
gamma  non-negative variance scaling parameter.
nthreads  number of threads for parallel execution via OpenMP.

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

## 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","*
","* number of rejections : ", sum(counter),"\n",)
mean2.ttest

"* total number of trials : ", niter,"\n",
"* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep="")

## End(Not run)

mean2.ttest

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

Description

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

\[
H_0 : \mu_x^2 \{=, \geq, \leq\} \mu_y^2 \quad \text{vs} \quad H_1 : \mu_x^2 \{\neq, <, >\} \mu_y^2
\]

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

Usage

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

Arguments

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

Value

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

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

Student (1908). “The Probable Error of a Mean.” Biometrika, 6(1), 1. ISSN 00063444.
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("\n* Example for 'mean2.ttest'\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"))
```

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

```r
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)
}

## 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

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

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.

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="\n\n"))

---

**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("* Example for 'meank.2019CPH'\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 univariate samples $X_1, \ldots, X_k$, it tests

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

Usage
```
meank.anova(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.
```

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
cat(paste("\n* Example for 'meank.anova'\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

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

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 \( \sigma^2_0 \).

Value

a (list) object of \( \texttt{S3} \) 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))

## Not run:
## empirical Type I error
niter = 1000
counter = rep(0,niter) # record p-values
for (i in 1:niter){
```
mvar1.LRT

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", "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)

---

mvar1.LRT

One-sample Simultaneous Likelihood Ratio Test of Mean and Variance

Description

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

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

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 \(\sigma_0^2\).

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))

## 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("Example for 'mvar1.LRT'
","*
","* 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

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

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

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)
mvar2.1930PN(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 \"Var mvar2.1930PN\"\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

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

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* Example for \rquote{mvar2.1976PL}\rquote{\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

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

using Muirhead's approximation for small-sample problem.

Usage

\[
\text{mvar2.1982Muirhead}(x, y)
\]

Arguments

- \( x \) : a length-\( n \) data vector.
- \( y \) : a length-\( m \) 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)
y = rnorm(10)
mvar2.1982Muirhead(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(mvar2.1982Muirhead(x,y)$p.value < 0.05, 1, 0)
}

## print the result
cat(paste("\n* Example for 'mvar2.1982Muirhead'\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)

---

**mvar2.2012ZXC**

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**

\[ \text{mvar2.2012ZXC}(x, y) \]

**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.
mvar2.LRT

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* Example for 'mvar2.2012ZXC'\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
\[
H_0 : \mu_x = \mu_y, \sigma^2_x = \sigma^2_y \quad vs \quad H_1 : \text{not } H_0
\]
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 \texttt{S3} class \texttt{htest} containing:

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

Examples

```r
## CRAN-purpose small example
x = rnorm(10)
y = rnorm(10)
mvar2.LRT(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.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=""))
## End(Not run)
```

---

\textit{norm.1965SW} \hspace{1cm} \textit{Univariate Test of Normality by Shapiro and Wilk (1965)}

Description

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

\[
H_0 : x \text{ is from normal distribution} \hspace{0.5cm} vs \hspace{0.5cm} H_1 : \text{not } H_0
\]

Usage

```r
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
```

---

**norm.1972SF**

*Univariate Test of Normality by Shapiro and Francia (1972)*

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

norm.1972SF(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
## 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

\[ \text{norm.1980JB}(x, \text{method} = \text{c("asymptotic", "MC"), nreps = 2000}) \]

Arguments

- \( x \) : a length-\( n \) data vector.
- \( \text{method} \) : method to compute \( p \)-value. Using initials is possible, "a" for asymptotic for example. Case insensitive.
- \( \text{nreps} \) : 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

```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

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

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.
- **method**: method to compute \( p \)-value. Using initials is possible, "a" for asymptotic for example.
- **nreps**: the number of Monte Carlo simulations to be run when method="MC".

**Value**

A (list) object of S3 class \( \text{htest} \) 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.1996AJB(x, method="a") # Asymptotics
test2 = norm.1996AJB(x, method="m") # Monte Carlo
```
norm.2008RJB

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.
- \( nreps \): the number of Monte Carlo simulations to be run when \( \text{method} = \text{"MC"} \).

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


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

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

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

## Not run:
## empirical Type I 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=""))

## End(Not run)

---

**Description**

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

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

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 'sim2.2018HN'\n","*
","* number of rejections : ", sum(counter),"\n",
","* total number of trials : ", niter,"\n",
","* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
```

unif.2017YMi

Multivariate Test of Uniformity based on Interpoint Distances by Yang and Modarres (2017)

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.

Usage

```r
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

```
## CRAN-purpose small example
smallX = matrix(rnorm(10*3),ncol=3)
unif.2017YMi(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.2017YMi(X, type="Q1")$p.value < 0.05, 1, 0)
  rec2[i] = ifelse(unif.2017YMi(X, type="Q2")$p.value < 0.05, 1, 0)
  rec3[i] = ifelse(unif.2017YMi(X, type="Q3")$p.value < 0.05, 1, 0)
}
## print the result
cat(paste("\n* Example for 'unif.2017YMi'\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=""))
```
unif.2017YMq

Multivariate Test of Uniformity based on Normal Quantiles by Yang and Modarres (2017)

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. 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* 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, ...)
```

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 = useknd(x, y, "meank.anova")
test2 = mean2.ttest(x, y)

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

---

**useknd**

*Apply k-sample tests for two multivariate samples*

**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**

```r
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

```r
## 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","\n","* number of rejections : ", sum(counter),"\n","* total number of trials : ", niter,"\n","* empirical Type 1 error : ",round(sum(counter/niter),5),"\n",sep=""))
```

---

**var1.chisq**

One-Sample Chi-Square Test for Variance

Description

Given an univariate sample $x$, it tests

$$H_0 : \sigma_x^2 \{=, \geq, \leq \} \sigma_0^2 \quad vs \quad H_1 : \sigma_x^2 \{\neq, <, > \} \sigma_0^2$$

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 \"Var 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=""))
```
Two-Sample F-Test for Variance

Description

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

$$ H_0 : \sigma_x^2 \in \{=, \geq, \leq\} \sigma_y^2 \quad \text{vs} \quad H_1 : \sigma_x^2 \notin \{=, <, >\} \sigma_y^2 $$

Usage

```r
var2.F(x, y, alternative = c("two.sided", "less", "greater"))
```

Arguments

- `x`: a length-$n$ data vector.
- `y`: a length-$m$ data vector.
- `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)
y = rnorm(10)
var2.F(x, y, alternative="g") # Ha : var(x) >= var(y)
var2.F(x, y, alternative="l") # Ha : var(x) <= var(y)
var2.F(x, y, alternative="t") # Ha : var(x) /= var(y)

## 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(var2.F(x,y)$p.value < 0.05, 1, 0)
}

## print the result
cat(paste("\nExample for 'var2.F'\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.1937Bartlett
Bartlett'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 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* Example for 'vark.1937Bartlett'
","\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

```r
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* number of rejections : ", sum(counter),"n",
"* total number of trials : ", niter,"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

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


Usage

\[ \text{vark.1974BF(dlist)} \]

Arguments

- \( \text{dlist} \): 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
  cov2.mxPBF, 9
  covk.2001Schott, 11
  covk.2007Schott, 12
* eqdist
  eqdist.2014BG, 13
* gof_normal
  norm.1965SW, 52
  norm.1972SF, 53
  norm.1980JB, 55
  norm.1996AJB, 56
  norm.2008RJB, 57
* gof_uniform
  unif.2017YMi, 60
  unif.2017YMq, 62
* mean_multivariate
  mean1.1931Hotelling, 15
  mean1.1958Dempster, 16
  mean1.1996BS, 17
  mean1.2008SD, 18
  mean2.1931Hotelling, 21
  mean2.1958Dempster, 22
  mean2.1965Yao, 23
  mean2.1980Johansen, 25
  mean2.1986NVM, 26
  mean2.1996BS, 27
  mean2.2004K, 29
  mean2.2008SD, 30
  mean2.2011JW, 31
  mean2.2014CLX, 33
  mean2.2014Thulin, 34
  mean2.mxPBF, 36
  meank.2007Schott, 38
  meank.2009ZX, 40
  meank.2019CPH, 41
* mean_univariate
  mean1.ttest, 19
  mean2.ttest, 37
  meank.anova, 43
* mvar
  mvar1.1998AS, 44
  mvar1.LRT, 45
  mvar2.1930PN, 46
  mvar2.1976PL, 47
  mvar2.1982Muirhead, 49
  mvar2.2012ZXC, 50
  mvar2.LRT, 51
* simtest
  sim1.2017Liu, 58
  sim2.2018HN, 59
* utility
  usek1d, 63
  useknd, 64
* variance
  var1.chisq, 65
  var2.F, 67
  vark.1937Bartlett, 68
  vark.1960Levene, 69
  vark.1974BF, 71

Bai and Saranadasa (1996), 38

cov1.2012Fisher, 3
  cov1.2015WL, 4
  cov2.2012LC, 5
  cov2.2013CLX, 7
  cov2.2015WL, 8
  cov2.mxPBF, 9
  covk.2001Schott, 11
  covk.2007Schott, 12
  eqdist.2014BG, 13
  mean1.1931Hotelling, 15
mean1.1958Dempster, 16
mean1.1996BS, 17
mean1.2008SD, 18
mean1.ttest, 19
mean2.1931Hotelling, 21
mean2.1958Dempster, 22
mean2.1965Yao, 23
mean2.1980Johansen, 25
mean2.1986NVM, 26
mean2.1996BS, 27
mean2.2004KY, 29
mean2.2008SD, 30
mean2.2011LJW, 31
mean2.2014CLX, 33
mean2.2014Thulin, 34
mean2.mxPBF, 36
mean2.ttest, 37
meank.2007Schott, 38
meank.2009ZX, 40
meank.2019CPH, 41
meank.anova, 43
mvar1.1998AS, 44
mvar1.LRT, 45
mvar2.1930PN, 46
mvar2.1976PL, 47
mvar2.1982Muirhead, 49
mvar2.2012ZXC, 50
mvar2.LRT, 51
norm.1965SW, 52
norm.1972SF, 53
norm.1980JB, 55
norm.1996AJB, 56
norm.2008RJB, 57
sim1.2017Liu, 58
sim2.2018HN, 59
unif.2017YMi, 60
unif.2017YMq, 62
usek1d, 63
useknd, 64
var1.chisq, 65
var2.F, 67
vark.1937Bartlett, 68
vark.1960Levene, 69
vark.1974BF, 71