Package 'SHT'

June 6, 2022

Type Package

Title Statistical Hypothesis Testing Toolbox

Version 0.1.7

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, pracma, flare

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.2.0

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 2022-06-06 20:10:22 UTC

R topics documented:

`cov1.2012Fisher` *One-sample Test for Covariance Matrix by Fisher (2012)*

Description

Given a multivariate sample X and hypothesized covariance matrix Σ_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.
`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

```
## 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",
"* 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 Σ_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

Wu T, Li P (2015). "Tests for High-Dimensional Covariance Matrices Using Random Matrix Projection." *arXiv:1511.01611 [stat]*.

Examples

```
## 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("\n* Example for 'cov1.2015WL'\n","*\n",
"* 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=""))
```

---

cov2.2012LC *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

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

using the procedure by Li and Chen (2012).

Usage

```
cov2.2012LC(X, Y, use.unbiased = TRUE)
```

Arguments

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

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

Li J, Chen SX (2012). "Two sample tests for high-dimensional covariance matrices." *The Annals of Statistics*, **40**(2), 908–940. ISSN 0090-5364.

Examples

```
## 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","*\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

Cai T, Liu W, Xia Y (2013). "Two-Sample Covariance Matrix Testing and Support Recovery in High-Dimensional and Sparse Settings." *Journal of the American Statistical Association*, **108**(501), 265–277. ISSN 0162-1459, 1537-274X.

Examples

```
## 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("\n* Example 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

```
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

Wu T, Li P (2015). "Tests for High-Dimensional Covariance Matrices Using Random Matrix Projection." *arXiv:1511.01611 [stat]*.

Examples

```
## 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","*\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.2001Schott` *Test for Homogeneity of Covariances by Schott (2001)*

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

Schott JR (2001). "Some tests for the equality of covariance matrices." *Journal of Statistical Planning and Inference*, **94**(1), 25–36. ISSN 03783758.

Examples

```
## 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)
  }

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

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

`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

```
## 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","*\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

```
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

Biswas M, Ghosh AK (2014). "A nonparametric two-sample test applicable to high dimensional data." *Journal of Multivariate Analysis*, **123**, 160–171. ISSN 0047259X.

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

## 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 1 ERROR COMPARISON \n","*\n",
"* Asymptotics : ", round(sum(pval.a/ntest),5),"\n",
"* Permutation : ", round(sum(pval.p/ntest),5),"\n",sep=""))
```

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

`mean1.1931Hotelling` *One-sample Hotelling's T-squared Test for Multivariate Mean*

Description

Given a multivariate sample X and hypothesized mean μ_0, it tests

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

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.
`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

Hotelling H (1931). "The Generalization of Student's Ratio." *The Annals of Mathematical Statistics*, **2**(3), 360–378. ISSN 0003-4851.

Examples

```
## 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)
}

## 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 μ_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

```
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 (1958). "A High Dimensional Two Sample Significance Test." *The Annals of Mathematical Statistics*, **29**(4), 995–1010. ISSN 0003-4851.

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

Examples

```
## 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 μ_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

```
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

Bai Z, Saranadasa H (1996). "HIGH DIMENSION: BY AN EXAMPLE OF A TWO SAMPLE PROBLEM." *Statistica Sinica*, **6**(2), 311–329. ISSN 10170405, 19968507.

Examples

```
## 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","*\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 μ_0, it tests

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

using the procedure by Srivastava and Du (2008).

Usage

```
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

Srivastava MS, Du M (2008). "A test for the mean vector with fewer observations than the dimension." *Journal of Multivariate Analysis*, **99**(3), 386–402. ISSN 0047259X.

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
cat(paste("\n* Example for 'mean1.2008SD'\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=""))
```

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

Description

Given an univariate sample x, it tests

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

using the procedure by Student (1908).

Usage

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

Arguments

x	a length-n data vector.
mu0	hypothesized mean μ_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.

Student (1908). "Probable Error of a Correlation Coefficient." *Biometrika*, **6**(2-3), 302–310. ISSN 0006-3444, 1464-3510.

Examples

```
## 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

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

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

Hotelling H (1931). "The Generalization of Student's Ratio." *The Annals of Mathematical Statistics*, **2**(3), 360–378. ISSN 0003-4851.

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","*\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

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

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 (1958). "A High Dimensional Two Sample Significance Test." *The Annals of Mathematical Statistics*, **29**(4), 995–1010. ISSN 0003-4851.

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

Examples

```
## 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("\n* Example for 'mean2.1958Dempster'\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=""))
```

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

Description

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

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

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 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

Yao Y (1965). "An Approximate Degrees of Freedom Solution to the Multivariate Behrens Fisher Problem." *Biometrika*, **52**(1/2), 139. ISSN 00063444.

Examples

```
## 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=""))
```

`mean2.1980Johansen` *Two-sample Test for Multivariate Means by Johansen (1980)*

Description

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

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

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

Usage

```
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

Johansen S (1980). "The Welch-James Approximation to the Distribution of the Residual Sum of Squares in a Weighted Linear Regression." *Biometrika*, **67**(1), 85. ISSN 00063444.

Examples

```
## 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){
```

```
  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

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

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:

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

Nel DG, Van Der Merwe CA (1986). "A solution to the multivariate behrens-fisher problem." *Communications in Statistics - Theory and Methods*, **15**(12), 3719–3735. ISSN 0361-0926, 1532-415X.

Examples

```
## 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","*\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

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

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

Bai Z, Saranadasa H (1996). "HIGH DIMENSION: BY AN EXAMPLE OF A TWO SAMPLE PROBLEM." *Statistica Sinica*, **6**(2), 311–329. ISSN 10170405, 19968507.

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",
"* total number of trials : ", niter,"\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

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

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

Krishnamoorthy K, Yu J (2004). "Modified Nel and Van der Merwe test for the multivariate Behrens–Fisher problem." *Statistics \& Probability Letters*, **66**(2), 161–169. ISSN 01677152.

Examples

```
## 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("\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

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

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

Srivastava MS, Du M (2008). "A test for the mean vector with fewer observations than the dimension." *Journal of Multivariate Analysis*, **99**(3), 386–402. ISSN 0047259X.

Examples

```
## 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=""))
```

---

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

```
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

Lopes ME, Jacob L, Wainwright MJ (2011). "A More Powerful Two-sample Test in High Dimensions Using Random Projection." In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, 1206–1214. ISBN 978-1-61839-599-3.

Examples

```
## 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
cat(paste("\n* Example for 'mean2.2011LJW'\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=""))
```

`mean2.2014CLX` *Two-sample Test for High-Dimensional Means by Cai, Liu, and Xia (2014)*

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 Cai, Liu, and Xia (2014) which is equivalent to test

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

for an inverse covariance (or precision) Ω. When Ω 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

```
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

Cai TT, Liu W, Xia Y (2014). "Two-sample test of high dimensional means under dependence." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **76**(2), 349–372. ISSN 13697412.

Examples

```
## 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'\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)
```

`mean2.2014Thulin` *Two-sample Test for Multivariate Means by Thulin (2014)*

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

```
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

Thulin M (2014). "A high-dimensional two-sample test for the mean using random subspaces." *Computational Statistics \& Data Analysis*, **74**, 26–38. ISSN 01679473.

Examples

```
## 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)

## 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 = 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

```
## 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",
```

```
"* 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

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

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

Usage

```
mean2.ttest(
  x,
  y,
  alternative = c("two.sided", "less", "greater"),
  paired = FALSE,
  var.equal = FALSE
)
```

Arguments

x	a length-n data vector.
y	a length-m data vector.
alternative	specifying the alternative hypothesis.
paired	a logical; whether consider two samples as paired.
var.equal	a logical; if FALSE, use Welch's correction.

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.

Student (1908). “Probable Error of a Correlation Coefficient.” *Biometrika*, **6**(2-3), 302–310. ISSN 0006-3444, 1464-3510.

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

Examples

```
## 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","*\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

```
## 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","*\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.2009ZX *Test for Equality of Means by Zhang and Xu (2009)*

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 Zhang and Xu (2009) by applying multivariate extension of Scheffe's method of transformation.

Usage

```
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

Zhang J, Xu J (2009). "On the k-sample Behrens-Fisher problem for high-dimensional data." *Science in China Series A: Mathematics*, **52**(6), 1285–1304. ISSN 1862-2763.

Examples

```
## 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","*\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

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

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

Usage

```
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

Cao M, Park J, He D (2019). “A test for the k sample Behrens–Fisher problem in high dimensional data.” *Journal of Statistical Planning and Inference*, **201**, 86–102. ISSN 03783758.

Examples

```
## 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)
```

`meank.anova` *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

```
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=""))
```

mvar1.1998AS *One-sample Simultaneous Test of Mean and Variance by Arnold and Shavelle (1998)*

Description

Given two univariate samples x and y, it tests

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

using asymptotic likelihood ratio test.

Usage

```
mvar1.1998AS(x, mu0 = 0, var0 = 1)
```

Arguments

x	a length-n data vector.
mu0	hypothesized mean μ_0.
var0	hypothesized variance σ_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.

References

Arnold BC, Shavelle RM (1998). "Joint Confidence Sets for the Mean and Variance of a Normal Distribution." *The American Statistician*, **52**(2), 133–140.

Examples

```
## CRAN-purpose small example
mvar1.1998AS(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(mvar1.1998AS(x)$p.value < 0.05, 1, 0)
}

## print the result
cat(paste("\n* Example 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

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

using likelihood ratio test.

Usage

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

Arguments

`x`	a length-n data vector.
`mu0`	hypothesized mean μ_0.
`var0`	hypothesized variance σ_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

```
## 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("\n* Example 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=""))

## 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

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

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

Usage

```
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

```
## 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

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

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

Perng SK, Littell RC (1976). "A Test of Equality of Two Normal Population Means and Variances." *Journal of the American Statistical Association*, **71**(356), 968–971. ISSN 0162-1459, 1537-274X.

Examples

```
## 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 'mvar2.1976PL'\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)
```

`mvar2.1982Muirhead` *Two-sample Simultaneous Test of Mean and Variance by Muirhead Approximation (1982)*

Description

Given two univariate samples x and y, it tests

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

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

Muirhead RJ (1982). *Aspects of multivariate statistical theory*, Wiley series in probability and mathematical statistics. Wiley, New York. ISBN 978-0-471-09442-5.

Examples

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

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

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

Zhang L, Xu X, Chen G (2012). "The Exact Likelihood Ratio Test for Equality of Two Normal Populations." *The American Statistician*, **66**(3), 180–184. ISSN 0003-1305, 1537-2731.

Examples

```
## 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","*\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

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

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 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

```
## 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","*\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)
```

---

norm.1965SW *Univariate Test of Normality by Shapiro and Wilk (1965)*

Description

Given an univariate sample x, it tests

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

using a test procedure by Shapiro and Wilk (1965). Actual computation of p-value is done via an approximation scheme by Royston (1992).

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

Shapiro SS, Wilk MB (1965). "An Analysis of Variance Test for Normality (Complete Samples)." *Biometrika*, **52**(3/4), 591. ISSN 00063444.

Royston P (1992). "Approximating the Shapiro-Wilk W-test for non-normality." *Statistics and Computing*, **2**(3), 117–119. ISSN 0960-3174, 1573-1375.

Examples

```
## 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

Shapiro SS, Francia RS (1972). "An Approximate Analysis of Variance Test for Normality." *Journal of the American Statistical Association*, **67**(337), 215–216. ISSN 0162-1459, 1537-274X.

Examples

```
## 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
```

norm.1980JB *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

```
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

Jarque CM, Bera AK (1980). "Efficient tests for normality, homoscedasticity and serial independence of regression residuals." *Economics Letters*, **6**(3), 255–259. ISSN 01651765.

Jarque CM, Bera AK (1987). "A Test for Normality of Observations and Regression Residuals." *International Statistical Review / Revue Internationale de Statistique*, **55**(2), 163. ISSN 03067734.

Examples

```
## 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
```

norm.1996AJB *Adjusted Jarque-Bera Test of Univariate Normality by Urzua (1996)*

Description

Given an univariate sample x, it tests

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

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

Usage

```
norm.1996AJB(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.
`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

Urzúa CM (1996). "On the correct use of omnibus tests for normality." *Economics Letters*, **53**(3), 247–251. ISSN 01651765.

Examples

```
## 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

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

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

Usage

```
norm.2008RJB(x, C1 = 6, C2 = 24, method = 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 $\alpha = 0.05$.
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 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

Gel YR, Gastwirth JL (2008). "A robust modification of the Jarque–Bera test of normality." *Economics Letters*, **99**(1), 30–32. ISSN 01651765.

Examples

```
## 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
```

`sim1.2017Liu` *One-sample Simultaneous Test of Mean and Covariance by Liu et al. (2017)*

Description

Given a multivariate sample X, hypothesized mean μ_0 and covariance Σ_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 procedure by Liu et al. (2017).

Usage

```
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

Liu Z, Liu B, Zheng S, Shi N (2017). “Simultaneous testing of mean vector and covariance matrix for high-dimensional data.” *Journal of Statistical Planning and Inference*, **188**, 82–93. ISSN 03783758.

Examples

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

`sim1.LRT` *One-sample Simultaneous Likelihood Ratio Test of Mean and Covariance*

Description

Given a multivariate sample X, hypothesized mean μ_0 and covariance Σ_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

```
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.

Examples

```
## CRAN-purpose small example
smallX = matrix(rnorm(10*3),ncol=3)
sim1.LRT(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(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
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)
```

`sim2.2018HN` *Two-sample Simultaneous Test of Means and Covariances by Hyodo and Nishiyama (2018)*

Description

Given a multivariate sample X, hypothesized mean μ_0 and covariance Σ_0, it tests

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

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

Usage

```
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

Hyodo M, Nishiyama T (2018). "A simultaneous testing of the mean vector and the covariance matrix among two populations for high-dimensional data." *TEST*, **27**(3), 680–699. ISSN 1133-0686, 1863-8260.

Examples

```
## 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","*\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

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

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

Yang M, Modarres R (2017). "Multivariate tests of uniformity." *Statistical Papers*, **58**(3), 627–639. ISSN 0932-5026, 1613-9798.

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","*\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

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

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

```
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

Yang M, Modarres R (2017). "Multivariate tests of uniformity." *Statistical Papers*, **58**(3), 627–639. ISSN 0932-5026, 1613-9798.

Examples

```
## 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

```
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

```
### 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 = usek1d(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=""))
```

---

usekind — *Apply k-sample tests for two multivariate samples*

Description

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

Usage

```
usekind(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","*\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

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

Arguments

x	a length-n data vector.
var0	hypothesized variance σ_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

Snedecor GW, Cochran WG (1996). *Statistical methods*, 8 ed., 7. print edition. Iowa State Univ. Press, Ames, Iowa. ISBN 978-0-8138-1561-9.

Examples

```
## 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=""))
```

`var2.F` *Two-Sample F-Test for Variance*

Description

Given two univariate samples x and y, it tests

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

.

Usage

```
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

Snedecor GW, Cochran WG (1996). *Statistical methods*, 8 ed., 7. print edition. Iowa State Univ. Press, Ames, Iowa. ISBN 978-0-8138-1561-9.

Examples

```
## 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("\n* Example 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

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

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

Bartlett MS (1937). "Properties of Sufficiency and Statistical Tests." *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, **160**(901), 268–282. ISSN 00804630.

Examples

```
## 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","*\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 $X_1 , \dots , X_k$, it tests

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

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

Levene H (1960). "Robust tests for equality of variances." In *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*, 278–292. Stanford University Press, Palo Alto, California.

Examples

```
## 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=""))
```

`vark.1974BF` *Brown-Forsythe Test for Homogeneity of Variance*

Description

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

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

using the procedure by Brown and Forsythe (1974).

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

Brown MB, Forsythe AB (1974). "Robust Tests for the Equality of Variances." *Journal of the American Statistical Association*, **69**(346), 364–367. ISSN 0162-1459, 1537-274X.

Examples

```
## 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