Package ‘highD2pop’

February 20, 2015

Type Package

Title Two-Sample Tests for Equality of Means in High Dimension

Version 1.0

Date 2012-11-02

Author Karl Gregory

Maintainer Karl Gregory <kgregory@mail.uni-mannheim.de>

Description Performs the generalized component test from Gregory et al (2015), as well as the tests from Chen and Qin (2010), Srivastava and Kubokawa (2013), and Cai, Liu, and Xia (2014) for equality of two population mean vectors when the length of the vectors exceeds the sample size.

License GPL (>= 2)

LazyLoad true

LazyData true

ZipData no

Depends fastclime

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-10-25 01:28:48

R topics documented:

highD2pop-package *Two-sample tests for equality of means in high dimension*

Description

This package provides functions for the tests from Gregory et al. (2015), Chen and Qin (2010), Srivastava and Kubokawa (2013), and Cai, Liu, and Xia (2014) for the equal means hypothesis in the high-dimensional, two-population setting. These are used to test

$$H_0 : \boldsymbol{\mu}_1 = \boldsymbol{\mu}_2$$

$$H_1 : \boldsymbol{\mu}_1 \neq \boldsymbol{\mu}_2$$

when the number of components in the mean vectors exceeds the sample size, that is in the *large-p-small-n* setting.

Details

Package:	highD2pop
Type:	Package
Version:	1.0
Date:	2012-11-02
License:	GPL (>=2)

The functions `GCT.test`, `ChenQin.test` and `SK.test`, `CLX.test.equalcov` and `CLX.test.unequalcov`, accept `n` by `p` and `m` by `p` data matrices with sample data from the first and second populations and return test statistics and p-values for the null hypothesis of equal means. The `build2popData` function simulates high-dimensional data in the two-population setting with specified sample sizes, numbers of components, covariance structure, etc., and the functions `GCT.sim`, `ChenQin.sim`, `SK.sim`, and `CLX.sim.Covtest` return test statistic values and p-values for lists of simulated data sets generated by `build2popData`. The `CLX.Covtest` function tests for equality of covariance matrices between the two populations with the test proposed in Cai, Liu, and Xia (2013). The `GCT.test.missing` is a version of the generalized component test which accomodates missing values and returns overall and componentwise missingness summaries. The functions `rdblepareto` and `rgammashift` generate realizations from heavy-tailed and skewed distributions under which the relative performance of the four tests is of interest.

Author(s)

Karl Gregory Maintainer: Karl Gregory <kgregory@mail.uni-mannheim.de>

References

Cai, T., Liu, W. and Luo, X. (2011). A constrained l-1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association* **106**, 594607.

Cai, T., Liu, W. and Xia, Y. (2013). Two-sample covariance matrix testing and support recovery. *Journal of the American Statistical Association* **108**, 265277.

Cai, T. T., Liu, W. and Xia, Y. (2014). Two-sample test of high dimensional means under dependence. *J. R. Statist. Soc. B.*

Chen, X. S. and Qin, Y.L. (2010). A two sample test for high dimensional data with applications to gene-set testing. *The Annals of Statistics*. **38**(2):808–835

Gregory, K., Carroll, R. J., Baladandayuthapani, V. and Lahiri, S. (2015). A two-sample test for equality of means in high dimension. *Journal of the American Statistician*, to appear

Hall, P. Jing, B. Y. and Lahiri, S. N. (1998). On the sampling window method for long-range dependent data. *Statistica Sinica* **8**,1189–1204

Srivastava, M. S. and Kubokawa, T. (2013). Tests for multivariate analysis of variance in high dimension under non-normality. *Journal of Multivariate Analysis* **115**, 204216.

Examples

```
data(chr1qseg)

impute<-function(x) {  x[which(is.na(x))] <- mean(x,na.rm=TRUE)
return(x)
}

X <- apply(chr1qseg$X,2,impute)
Y <- apply(chr1qseg$Y,2,impute)

## on imputed data with no missing values:

ChenQin.test(X,Y)
GCT.test(X,Y,r=20,smoother="parzen")
SK.test(X,Y)

## on raw data with missing values:

GCT.test.missing(chr1qseg$X,chr1qseg$Y,r=20,smoother="parzen")
```

build2popData *Data simulator for the high-dimensional two-sample setting*

Description

Generates data from two populations with user-specified mean vectors, covariance structure, sample sizes, and dimension of each observation.

Usage

```
build2popData(n, m, p, muX, muY, dep, commoncov, VarScaleY,
 ARMAparms, LRparm,S = 1, innov = function(n, ...) rnorm(n, 0, 1),
 heteroscedastic = FALSE, het.diag)
```

Arguments

n number of observations in sample one.

m number of observations in sample two.

p number of components in each observation.

muX p by 1 vector of component means for population one.

muY p by 1 vector of component means for population two.

dep dependence structure among the p components for both populations. Choices are 'IND', 'ARMA', and 'LR'.

commoncov a logical indicating whether populations one and two will have equal covariance matrices. If FALSE, then the innovations used in generating the population two data will be scaled by the square root of the value specifed in VarScaleY.

VarScaleY constant by which innovations are scaled in generating observations for sample two when commoncov=FALSE.

ARMAparms a list of the form list(coefs=list(ma= , ar=)) specifying the AR and MA parameters to be used when dep='ARMA'.

LRparm value of the LR dependence parameter to be used when dep='LR'.

S the number of data sets to simulate.

innov a function used to generate the innovations, such as innov=function(n,...) rnorm(n,0,1).

heteroscedastic

a logical indicating whether the components will be scaled by the entries in the diagonal matrix specified by het.diag=.

het.diag a p by p diagonal matrix, where the entries on the diagonal will be used to scale the component standard deviations.

Value

A list of length S of lists, each containing

X the n by p data matrix for sample one.

Y the m by p data matrix for sample two.

n the number of observations in sample one.

m the number of observation in sample two.

p the number of components in each observation.

muX the mean vector for population one.

muY the mean vector for population two.

dep the dependence structure chosen for dep.

commoncov the value of commoncov used in the function call.

VarScaleY the scalar by which the variance of the population two data is scaled.

ARMAparms the list containing the specified ARMA parameters.

LRparm the long-range dependence parameter.

S the number of simulated data sets.

innov the function chosen to generate the innovations.

heteroscedastic
logical indicating whether heteroscedastic=TRUE in the function call.

het.diag the value of het.diag in the function call.

Author(s)

Karl Gregory <kgregory@mail.uni-mannheim.de>, http://www.stat.tamu.edu/~kbgregory.

References

Hall, P. Jing, B. Y. and Lahiri, S. N. (1998). On the sampling window method for long-range dependent data. *Statistica Sinica* **8**,1189–1204

Examples

```
## Not run:
DATA <-build2popData(
n = 15,
m = 20,
p = 500,
muX = rep(0,500),
muY = rep(0,500),
commoncov = FALSE,
VarscaleY = 2,
dep = "ARMA",
ARMAparms = list(coefs=list(ma=c(.2,.3) , ar=c(.4,-.1))),
LRparm = .75,
S = 25,
innov = function(n,...) rnorm(n,0,1),
heteroscedastic=TRUE,
het.diag = diag(.1 + rexp(500,1/2))
)

## End(Not run)
```

build2popData_muYafunction

Data simulator for the high-dimensional two-sample setting

Description

Generates data from two populations with user-specified mean vectors, that of population two a function which returns the vector of means, covariance structure, sample sizes, and dimension of each observation.

Usage

```
build2popData_muYafunction(n, m, p, muX, muY, dep, commoncov, VarScaleY,
 ARMAparms, LRparm,S = 1, innov = function(n, ...) rnorm(n, 0, 1),
 heteroscedastic = FALSE, het.diag)
```

Arguments

`n` number of observations in sample one.

`m` number of observations in sample two.

`p` number of components in each observation.

`muX` `p` by 1 vector of component means for population one.

`muY` `p` by 1 a function returning a vector of length p which whill serve as the vector of component means for population two.

`dep` dependence structure among the `p` components for both populations. Choices are `'IND'`, `'ARMA'`, and `'LR'`.

`commoncov` a logical indicating whether populations one and two will have equal covariance matrices. If `FALSE`, then the innovations used in generating the population two data will be scaled by the square root of the value specifed in `VarScaleY`.

`VarScaleY` constant by which innovations are scaled in generating observations for sample two when `commoncov=FALSE`.

`ARMAparms` a list of the form `list(coefs=list(ma= , ar=))` specifying the AR and MA parameters to be used when `dep='ARMA'`.

`LRparm` value of the LR dependence parameter to be used when `dep='LR'`.

`S` the number of data sets to simulate.

`innov` a function used to generate the innovations, such as `innov=function(n,...) rnorm(n,0,1)`.

`heteroscedastic`
a logical indicating whether the components will be scaled by the entries in the diagonal matrix specified by `het.diag=`.

`het.diag` a `p` by `p` diagonal matrix, where the entries on the diagonal will be used to scale the component standard deviations.

Value

A list of length `S` of lists, each containing

`X` the `n` by `p` data matrix for sample one.

`Y` the `m` by `p` data matrix for sample two.

`n` the number of observations in sample one.

`m` the number of observation in sample two.

`p` the number of components in each observation.

`muX` the mean vector for population one.

`muY` the mean vector for population two.

`dep` the dependence structure chosen for `dep`.

`commoncov` the value of `commoncov` used in the function call.

`VarScaleY` the scalar by which the variance of the population two data is scaled.

`ARMAparms` the list containing the specified ARMA parameters.

`LRparm` the long-range dependence parameter.

`S` the number of simulated data sets.

`innov` the function chosen to generate the innovations.

`heteroscedastic`
logical indicating whether `heteroscedastic=TRUE` in the function call.

`het.diag` the value of `het.diag` in the function call.

Author(s)

Karl Gregory <`kgregory@mail.uni-mannheim.de`>, `http://www.stat.tamu.edu/~kbgregory`.

References

Hall, P. Jing, B. Y. and Lahiri, S. N. (1998). On the sampling window method for long-range dependent data. *Statistica Sinica* **8**,1189–1204

Examples

```
## Not run:
DATA <-build2popData(
n = 15,
m = 20,
p = 500,
muX = rep(0,500),
muY = rep(0,500),
commoncov = FALSE,
VarscaleY = 2,
dep = "ARMA",
ARMAparms = list(coefs=list(ma=c(.2,.3) , ar=c(.4,-.1))),
LRparm = .75,
S = 25,
innov = function(n,...) rnorm(n,0,1),
heteroscedastic=TRUE,
het.diag = diag(.1 + rexp(500,1/2))
)

## End(Not run)
```

center *The centering value for the test statistic*

Description

This is a helper function for the generalized component test from Gregory et al. (2014) which computes and returns the value $1+\hat{c}_{jn}/n+\hat{d}_{jn}/n^2$ for a component j. The `GCT.test` and `GCT.sim` average these values across $j = 1, \ldots, p$ to obtain the centering $1+\hat{a}_n/n+\hat{b}_n/n^2$ for the generalized component test statistic.

Usage

```
center(xy,n,m,ntoorderminus=2)
```

Arguments

`xy` a vector of length `n + m` containing the concatenation of the `n` realizations of a given component from the first sample and the `m` realizations of the same component from the second sample.

`n` the number of entries in `xy` belonging to the first sample.

`m` the number of entries in `xy` belonging to the second sample.

`ntoorderminus` the order of terms retained in $1+\hat{c}_{jn}/n+\hat{d}_{jn}/n^2$. If `ntoorderminus=0`, then 1 is returned. If `ntoorderminus=1` then $1+\hat{c}_{jn}/n$ is returned. If `ntoorderminus=2` then $1+\hat{c}_{jn}/n+\hat{d}_{jn}/n^2$ is returned.

Value

A scalar equal to 1, $1+\hat{c}_{jn}/n$, or $1+\hat{c}_{jn}/n+\hat{d}_{jn}/n^2$, respectively, for `ntoorderminus=0,1,2`.

Author(s)

Karl Gregory <`kgregory@mail.uni-mannheim.de`>, `http://www.stat.tamu.edu/~kbgregory`.

References

Gregory, K., Carroll, R. J., Baladandayuthapani, V. and Lahiri, S. (2015). A two-sample test for equality of means in high dimension. *Journal of the American Statistician*, to appear

---

`ChenQin.sim` *Chen Qin Simulator*

Description

Performs the test from Chen and Qin (2010) on multiple data sets generated by `build2popData`.

Usage

```
ChenQin.sim(DATA)
```

Arguments

`DATA`	an object returned by `build2popData`.

Value

A data frame with columns

`ChQvalues`	values of the test statistic.
`pvalues`	the p values.

Author(s)

Karl Gregory <kgregory@mail.uni-mannheim.de>, `http://www.stat.tamu.edu/~kbgregory`.

References

Chen, X. S. and Qin, Y.L. (2010). A two sample test for high dimensional data with applications to gene-set testing. *The Annals of Statistics*. **38**(2):808–835

See Also

`ChenQin.test`

Examples

```
## Not run:

DATA <-build2popData(
n = 15,
m = 20,
p = 500,
muX = rep(0,500),
muY = rep(0,500),
commoncov = FALSE,
VarScaleY = 1,
dep = "ARMA",
ARMAparms = list(coefs=list(ma=c(.2,.3) , ar=c(.4,-.1))),
LRparm = .75,
S = 25,
innov = function(n,...) rnorm(n,0,1),
heteroscedastic=TRUE,
het.diag = diag(.1 + rexp(500,1/2))
)

ChenQin.sim(DATA)


## End(Not run)
```

`ChenQin.test` *Chen's and Qin's test for equality of two mean vectors*

Description

Performs the test from Chen and Qin (2010) for the equality of two `p` by `1` population mean vectors given samples of sizes `n` and `m`.

Usage

```
ChenQin.test(X, Y)
```

Arguments

`X` the `n` by `p` data matrix for sample one.

`Y` the `m` by `p` data matrix for sample two.

Value

`ChQ` the value of the test statistic.

`pvalue` the two-sided p-value for the test statistic.

Author(s)

Karl Gregory <kgregory@mail.uni-mannheim.de>, http://www.stat.tamu.edu/~kbgregory.

References

Chen, X. S. and Qin, Y.L. (2010). A two sample test for high dimensional data with applications to gene-set testing. *The Annals of Statistics*. **38**(2):808–835

Examples

```
## Not run:

data(chr1qseg)

impute <- function(x)
{
x[which(is.na(x))] <- mean(x,na.rm=TRUE)
return(x)
}

X <- apply(chr1qseg$X,2,impute)
Y <- apply(chr1qseg$Y,2,impute)

ChenQin.test(X,Y)


## End(Not run)
```

---

chr1qseg *Copy number data along a segment of the q arm of chromosome 1*

Description

.Rdata object containing copy number probes from TCGA at 400 locations (given in 'loc', which is number of base pairs from end of chromosome arm) on the q arm of chromosome 1 for 92 long term survivors 'X' and 138 short term survivors 'Y' of glioblastoma multiforme. Around 3% of the values are missing.

Usage

```
chr1qseg
```

Format

A list containing `X`, the `n=92` by `p=400` matrix of copy number measurements for the `n=92` long-term survivors, `Y`, the `m=138` by `p=400` matrix of copy number measurements for the `m=138` short-term survivors, and `loc`, which gives the locations of the `400` measurements in number of base pairs from the end of the q arm of chromosome 1.

CLX.Covtest *Cai, Liu, and Xia test for equality of two mean vectors*

Description

Performs the test from Cai, Liu, and Xia (2013) for the equality of two population covariance matrices.

Usage

```
CLX.Covtest(X, Y)
```

Arguments

`X`	the `n` by `p` data matrix for sample one.
`Y`	the `m` by `p` data matrix for sample two.

Value

`TSvalue`	the value of the test statistic.
`pvalue`	the p-value for the test statistic.

Author(s)

Karl Gregory `<kgregory@mail.uni-mannheim.de>`, `http://www.stat.tamu.edu/~kbgregory`.

References

T. Tony Cai, Weidong Liu, and Yin Xia. 2013 "Two-sample covariance matrix testing and support recovery in high-dimensional and sparse settings" *Journal of The American Statistical Association*. **108**, 265–277

See Also

`CLX.sim.Covtest`

Examples

```
## Not run:

data(chr1qseg)

impute <- function(x)
{
x[which(is.na(x))] <- mean(x,na.rm=TRUE)
return(x)
}
```

```
X <- apply(chr1qseg$X,2,impute)
Y <- apply(chr1qseg$Y,2,impute)

CLX.test.equalcov(X,Y)


## End(Not run)
```

`CLX.sim.Covtest` *Cai, Liu, and Xia equal means test simulator*

Description

Performs the test in Cai, Liu, and Xia (2014) for testing equality of two mean vectors on multiple data sets generated by `build2popData`, performing on each data set the test for equal covariance matrices found in Cai, Liu, and Xia (2013).

Usage

```
CLX.sim.Covtest(DATA)
```

Arguments

`DATA`	an object returned by `build2popData`.

Value

A data frame with columns

`TSvalues`	values of the test statistic.
`pvalues`	the p values.

Author(s)

Karl Gregory <`kgregory@mail.uni-mannheim.de`>, `http://www.stat.tamu.edu/~kbgregory`.

References

Cai, T., Liu, W. and Xia, Y. (2013). Two-sample covariance matrix testing and support recovery. *Journal of the American Statistical Association* **108**, 265277.

Cai, T. T., Liu, W. and Xia, Y. (2014). Two-sample test of high dimensional means under dependence. *J. R. Statist. Soc. B.*

See Also

`CLX.Covtest`

Examples

```
## Not run:
DATA <-build2popData(
n = 15,
m = 20,
p = 500,
muX = rep(0,500),
muY = rep(0,500),
commoncov = FALSE,
VarScaleY = 1,
dep = "ARMA",
ARMAparms = list(coefs=list(ma=c(.2,.3) , ar=c(.4,-.1))),
LRparm = .75,
S = 25,
innov = function(n,...) rnorm(n,0,1),
heteroscedastic=TRUE,
het.diag = diag(.1 + rexp(500,1/2))
)
CLX.sim.Covtest(DATA)


## End(Not run)
```

CLX.sim.equalcov *Cai, Liu, and Xia equal means test simulator under equal covariances*

Description

Performs the test from Cai, Liu, and Xia (2014) for testing equality of two mean vectors on multiple data sets generated by build2popData when the population covariances matrices can be assumed equal.

Usage

```
CLX.sim.equalcov(DATA)
```

Arguments

DATA an object returned by build2popData.

Value

A data frame with columns

TSvalues values of the test statistic.

pvalues the p values.

Author(s)

Karl Gregory <kgregory@mail.uni-mannheim.de>, http://www.stat.tamu.edu/~kbgregory.

References

Cai, T. T., Liu, W. & Xia, Y. (2014). Two-sample test of high dimensional means under dependence. *J. R. Statist. Soc. B.*

See Also

CLX.test.equalcov

Examples

```
## Not run:

DATA <-build2popData(
n = 15,
m = 20,
p = 500,
muX = rep(0,500),
muY = rep(0,500),
commoncov = FALSE,
VarScaleY = 1,
dep = "ARMA",
ARMAparms = list(coefs=list(ma=c(.2,.3) , ar=c(.4,-.1))),
LRparm = .75,
S = 25,
innov = function(n,...) rnorm(n,0,1),
heteroscedastic=TRUE,
het.diag = diag(.1 + rexp(500,1/2))
)

CLX.sim.equalcov(DATA)


## End(Not run)
```

CLX.sim.unequalcov *Cai, Liu, and Xia equal means test simulator under unequal covariances*

Description

Performs the test in Cai, Liu, and Xia (2014) for testing equality of two mean vectors on multiple data sets generated by build2popData when the population covariances matrices are assumed to be unequal.

Usage

```
CLX.sim.unequalcov(DATA)
```

Arguments

DATA	an object returned by build2popData.

Value

A data frame with columns

TSvalues	values of the test statistic.
pvalues	the p values.

Author(s)

Karl Gregory <kgregory@mail.uni-mannheim.de>, http://www.stat.tamu.edu/~kbgregory.

References

Cai, T. T., Liu, W. & Xia, Y. (2014). Two-sample test of high dimensional means under dependence. *J. R. Statist. Soc. B.*

See Also

CLX.test.unequalcov

Examples

```
## Not run:

DATA <-build2popData(
n = 15,
m = 20,
p = 500,
muX = rep(0,500),
muY = rep(0,500),
commoncov = FALSE,
VarScaleY = 1,
dep = "ARMA",
ARMAparms = list(coefs=list(ma=c(.2,.3) , ar=c(.4,-.1))),
LRparm = .75,
S = 25,
innov = function(n,...) rnorm(n,0,1),
heteroscedastic=TRUE,
het.diag = diag(.1 + rexp(500,1/2))
)

CLX.sim.unequalcov(DATA)
```

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

`CLX.test.equalcov` *Cai, Liu, and Xia equal means test under equal covariances*

Description

Performs the test in Cai, Liu, and Xia (2014) for the equality of two `p` by `1` population mean vectors given samples of sizes `n` and `m` when the popoulation covariance matrices can be assumed equal.

Usage

```
CLX.test.equalcov(X, Y)
```

Arguments

`X`	the `n` by `p` data matrix for sample one.
`Y`	the `m` by `p` data matrix for sample two.

Value

`TSvalue`	the value of the test statistic.
`pvalue`	the two-sided p-value for the test statistic.

Author(s)

Karl Gregory <`kgregory@mail.uni-mannheim.de`>, `http://www.stat.tamu.edu/~kbgregory`.

References

Cai, T. T., Liu, W. & Xia, Y. (2014). Two-sample test of high dimensional means under dependence. *J. R. Statist. Soc. B.*

See Also

`CLX.sim.equalcov`

Examples

```
## Not run:

data(chr1qseg)

impute <- function(x)
{
x[which(is.na(x))] <- mean(x,na.rm=TRUE)
return(x)
}

X <- apply(chr1qseg$X,2,impute)
Y <- apply(chr1qseg$Y,2,impute)

CLX.test.equalcov(X,Y)


## End(Not run)
```

`CLX.test.unequalcov` *Cai, Liu, and Xia equal means test under unequal covariances*

Description

Performs the test in Cai, Liu, and Xia (2014) for the equality of two `p` by `1` population mean vectors given samples of sizes `n` and `m` when the popoulation covariance matrices are assumed to be unequal.

Usage

```
CLX.test.unequalcov(X, Y)
```

Arguments

`X` the `n` by `p` data matrix for sample one.

`Y` the `m` by `p` data matrix for sample two.

Value

`TSvalue` the value of the test statistic.

`pvalue` the two-sided p-value for the test statistic.

Author(s)

Karl Gregory <`kgregory@mail.uni-mannheim.de`>, `http://www.stat.tamu.edu/~kbgregory`.

References

Cai, T. T., Liu, W. & Xia, Y. (2014). Two-sample test of high dimensional means under dependence. *J. R. Statist. Soc. B.*

See Also

`CLX.sim.equalcov`

Examples

```
## Not run:

data(chr1qseg)

impute <- function(x)
{
x[which(is.na(x))] <- mean(x,na.rm=TRUE)
return(x)
}

X <- apply(chr1qseg$X,2,impute)
Y <- apply(chr1qseg$Y,2,impute)

CLX.test.unequalcov(X,Y)


## End(Not run)
```

GCT.sim *Generalized component test simulator*

Description

Performs the generalized component test from Gregory et al. (2014) on multiple data sets generated by build2popData.

Usage

```
GCT.sim(DATA, r, smoother = "parzen", ntoorderminus = 2)
```

Arguments

DATA an object returned by build2popData which is a list of S data sets.

r the lag window size for variance estimation.

smoother the lag window used in the variance estimation. Possible values are "parzen" and "trapezoid".

ntoorderminus a value of 0,1, or 2 such that the centering constant will retain terms of order n^(-ntoorderminus). Enter 0 for the moderate-p GCT, and enter 2 for the large-p GCT. A value of 1 may be entered to retain only terms which are O(1/n), appropriate for a size of p between moderate and large.

Value

A data frame with columns

TSvalues the values of the unstudentized test statistic.

`pvalues` the p-values.

`smoothtype` the choice of the lag window used in variance estimation.

`T` the values of the studentized test statistic.

`cent` the values of the centering constant.

`var` the values of the estimated variance of the test statistic.

Author(s)

Karl Gregory <kgregory@mail.uni-mannheim.de>, http://www.stat.tamu.edu/~kbgregory.

References

Gregory, K., Carroll, R. J., Baladandayuthapani, V. and Lahiri, S. (2015). A two-sample test for equality of means in high dimension. *Journal of the American Statistician*, to appear

See Also

`GCT.test`

Examples

```
## Not run:

DATA <-build2popData(
n = 15,
m = 20,
p = 500,
muX = rep(0,500),
muY = rep(0,500),
commoncov = FALSE,
VarScaleY = 1,
dep = "ARMA",
ARMAparms = list(coefs=list(ma=c(.2,.3) , ar=c(.4,-.1))),
LRparm = .75,
S = 25,
innov = function(n,...) rnorm(n,0,1),
heteroscedastic=TRUE,
het.diag = diag(.1 + rexp(500,1/2))
)

GCT.sim(DATA,r=20,smoother="parzen")


## End(Not run)
```

GCT.test *Generalized component test*

Description

Performs the generalized component test from Gregory et al. (2014) for the equality of two `p` by `1` population mean vectors given samples of sizes `n` and `m`.

Usage

```
GCT.test(X, Y, r, smoother = "parzen", ntoorderminus = 2)
```

Arguments

`X` the `n` by `p` data matrix for sample one.

`Y` the `m` by `p` data matrix for sample two.

`r` the lag window size for variance estimation.

`smoother` the lag window used in the variance estimation. Possible values are `"parzen"` and `"trapezoid"`.

`ntoorderminus` a value of `0,1`, or `2` such that the centering constant will retain terms of order `n^(-ntoorderminus)`. Enter `0` for the moderate-p GCT, and enter `2` for the large-p GCT. A value of `1` may be entered to retain only terms which are O(1/n), appropriate for a size of `p` between moderate and large.

Value

`TSvalue` the unstudentized test statistic.

`center` the centering constant for studentizing the test statistic.

`var` the estimated variance of the unstudentized test statistic.

`T` the studentized test statistic.

`smoother` the choice of smoother used.

`pvalue` the p-value.

Author(s)

Karl Gregory <`kgregory@mail.uni-mannheim.de`>, `http://www.stat.tamu.edu/~kbgregory`.

References

Gregory, K., Carroll, R. J., Baladandayuthapani, V. and Lahiri, S. (2015). A two-sample test for equality of means in high dimension. *Journal of the American Statistician*, to appear

See Also

`GCT.test.missing`

Examples

```
## Not run:
data(chr1qseg)

impute <- function(x)
{
x[which(is.na(x))] <- mean(x,na.rm=TRUE)
return(x)
}

X <- apply(chr1qseg$X,2,impute)
Y <- apply(chr1qseg$Y,2,impute)


GCT.test(X,Y)


## End(Not run)
```

`GCT.test.missing` *Generalized component test for missing data*

Description

Performs the generalized component test from Gregory et al. (2014) for the equality of two `p` by `1` population mean vectors given samples of sizes `n` and `m` when there are missing data.

Usage

```
GCT.test.missing(X, Y, r, smoother = "parzen", ntoorderminus = 2)
```

Arguments

`X` the `n` by `p` data matrix for sample one.

`Y` the `m` by `p` data matrix for sample two.

`r` the lag window size for variance estimation.

`smoother` the lag window used in the variance estimation. Possible values are `"parzen"` and `"trapezoid"`.

`ntoorderminus` a value of `0,1`, or `2` such that the centering constant will retain terms of order `n^(-ntoorderminus)`. Enter `0` for the moderate-p GCT, and enter `2` for the large-p GCT. A value of `1` may be entered to retain only terms which are O(1/n), appropriate for a size of `p` between moderate and large.

Value

`TSvalue` the unstudentized test statistic.

`center` the centering constant for studentizing the test statistic.

`var` the estimated variance of the unstudentized test statistic.

`T` the studentized test statistic.

`smoother` the choice of smoother used.

`pvalue` the pvalue.

`overallpctmiss` the overall proportion of values that are missing.

`pctmissperX` a vector of length `p` containing the proportion of missing values per component in sample one.

`pctmissperY` a vector of length `p` containing the proportion of missing values per component in sample two.

Author(s)

Karl Gregory <kgregory@mail.uni-mannheim.de>, http://www.stat.tamu.edu/~kbgregory.

References

Gregory, K., Carroll, R. J., Baladandayuthapani, V. and Lahiri, S. (2015). A two-sample test for equality of means in high dimension. *Journal of the American Statistician*, to appear

See Also

`GCT.test`

Examples

```
data(chr1qseg)

X <- chr1qseg$X
Y <- chr1qseg$Y

GCT.test.missing(X,Y,r=20,smoother="parzen")
```

`rdblepareto` *The double pareto distribution*

Description

Random number generation from the double Pareto distribution with parameters `shape` and `scale`.

Usage

```
rdblepareto(n, shape, scale)
```

Arguments

n	number of observations to generate.
shape	shape parameter, must be strictly greater than zero.
scale	scale parameter, must be strictly greater than zero.

Value

a vector of length n of realizations from the double Pareto distribution with shape parameter shape and scale parameter scale.

Author(s)

Karl Gregory <kgregory@mail.uni-mannheim.de>, http://www.stat.tamu.edu/~kbgregory.

Examples

```
## The function is currently defined as
function (n, shape, scale)
{
    u <- runif(n, 0, 1)
    x <- scale * (1 - u)^(-1/shape) - scale
    y <- sample(c(-1, 1), n, replace = TRUE) * x
    return(y)
  }


x <- rdblepareto(500,1.5,1)

plot(density(x))
```

rgammashift *The centered gamma distribution*

Description

Random generation for the Gamma distribution with parameters shape and scale centered so as to have mean zero.

Usage

```
rgammashift(n, shape, scale)
```

Arguments

`n`	number of observations to generate.
`shape`	shape parameter, must be strictly greater than zero.
`scale`	scale parameter, must be strictly greater than zero.

Value

a vector of length `n` of realizations from the centered gamma distribution with shape parameter `shape` and scale parameter `scale`.

Author(s)

Karl Gregory <`kgregory@mail.uni-mannheim.de`>, `http://www.stat.tamu.edu/~kbgregory`.

Examples

```
## The function is currently defined as
function (n, shape, scale)
{
    x <- rgamma(n = n, shape = shape, scale = scale) - shape*scale
    return(x)
}


x <- rgammashift(500,4,2)

plot(density(x))
```

`SK.sim` *Srivastava and Kubokawa Simulator*

Description

Performs the test from Srivastava and Kubokawa (2013) on multiple data sets generated by `build2popData`.

Usage

```
SK.sim(DATA)
```

Arguments

`DATA`	an object returned by `build2popData`.

Value

A data frame with columns

`TSvalues`	values of the test statistic.
`pvalues`	the p values.

Author(s)

Karl Gregory <kgregory@mail.uni-mannheim.de>, http://www.stat.tamu.edu/~kbgregory.

References

Srivastava, M. S. and Kubokawa, T. (2013). Tests for multivariate analysis of variance in high dimension under non-normality. *Journal of Multivariate Analysis* **115**, 204216.

See Also

`SK.test`

Examples

```
## Not run:
DATA <-build2popData(
n = 15,
m = 20,
p = 500,
muX = rep(0,500),
muY = rep(0,500),
commoncov = FALSE,
VarScaleY = 1,
dep = "ARMA",
ARMAparms = list(coefs=list(ma=c(.2,.3) , ar=c(.4,-.1))),
LRparm = .75,
S = 25,
innov = function(n,...) rnorm(n,0,1),
heteroscedastic=TRUE,
het.diag = diag(.1 + rexp(500,1/2))
)

SK.sim(DATA)


## End(Not run)
```

SK.test *Srivastava and Kubokawa test for equal means*

Description

Performs the test from Srivastava and Kubokawa (2013) for the equality of two p by 1 population mean vectors given samples of sizes n and m.

Usage

```
SK.test(X, Y)
```

Arguments

X	the n by p data matrix for sample one.
Y	the m by p data matrix for sample two.

Value

TSvalue	the value of the test statistic.
pvalue	the two-sided p-value for the test statistic.

Author(s)

Karl Gregory <kgregory@mail.uni-mannheim.de>, http://www.stat.tamu.edu/~kbgregory.

References

Srivastava, M. S. and Kubokawa, T. (2013). Tests for multivariate analysis of variance in high dimension under non-normality. *Journal of Multivariate Analysis* **115**, 204216.

See Also

SK.sim

Examples

```
## Not run:

data(chr1qseg)

impute <- function(x)
{
x[which(is.na(x))] <- mean(x,na.rm=TRUE)
return(x)
}

X <- apply(chr1qseg$X,2,impute)
```

```
Y <- apply(chr1qseg$Y,2,impute)

SK.test(X,Y)


## End(Not run)
```

Index