Package ‘ICSNP’

September 23, 2015

Type Package
Title Tools for Multivariate Nonparametrics
Version 1.1-0
Date 2015-09-22
Author Klaus Nordhausen, Seija Sirkia, Hannu Oja, David E. Tyler
Maintainer Klaus Nordhausen <klaus.nordhausen@utu.fi>
Depends R (>= 2.4.0), mvtnorm, ICS
Description Tools for multivariate nonparametrics, as location tests based on marginal ranks, spatial median and spatial signs computation, Hotelling's T-test, estimates of shape are implemented.
License GPL (>= 2)
Encoding latin1
NeedsCompilation yes
Repository CRAN
Date/Publication 2015-09-23 02:50:10

R topics documented:

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICSNP-package</td>
<td>2</td>
</tr>
<tr>
<td>duembgen.shape</td>
<td>3</td>
</tr>
<tr>
<td>duembgen.shape.wt</td>
<td>4</td>
</tr>
<tr>
<td>hl.loc</td>
<td>5</td>
</tr>
<tr>
<td>HotellingsT2</td>
<td>6</td>
</tr>
<tr>
<td>HP.loc.test</td>
<td>8</td>
</tr>
<tr>
<td>HP1.shape</td>
<td>10</td>
</tr>
<tr>
<td>HR_Mest</td>
<td>11</td>
</tr>
<tr>
<td>ind.cctest</td>
<td>12</td>
</tr>
<tr>
<td>ind.ictest</td>
<td>14</td>
</tr>
<tr>
<td>LASERI</td>
<td>15</td>
</tr>
<tr>
<td>pair.diff</td>
<td>17</td>
</tr>
<tr>
<td>pair.prod</td>
<td>18</td>
</tr>
<tr>
<td>pair.sum</td>
<td>19</td>
</tr>
<tr>
<td>pulmonary</td>
<td>20</td>
</tr>
</tbody>
</table>
ICSNP-package

### Description

This package contains functions and tools for multivariate nonparametric tests and estimation.

### Details

<table>
<thead>
<tr>
<th>Package:</th>
<th>ICSNP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>Package</td>
</tr>
<tr>
<td>Version:</td>
<td>1.1-0</td>
</tr>
<tr>
<td>Date:</td>
<td>2015-09-22</td>
</tr>
<tr>
<td>License:</td>
<td>GPL (&gt;= 2)</td>
</tr>
</tbody>
</table>

This package contains tools for nonparametric multivariate analysis, including the estimation of location and shape as well as some tests for location and independence. Shape matrices from this package can be used as one of the scatter matrices needed in the package ICS whereas the tests of this package can be used for testing in the framework of invariant coordinates or independent components obtained from the package ICS. The parametric Hotelling’s T test serves as a reference for the nonparametric location tests.

### Author(s)

Klaus Nordhausen, Seija Sirkiä, Hannu Oja and David E. Tyler

Maintainer: Klaus Nordhausen, <klaus.nordhausen@utu.fi>

### See Also

ICS
**duembgen.shape**

---

**Duembgen’s Shape Matrix**

---

**Description**

Iterative algorithm to estimate Dümbgen’s shape matrix.

**Usage**

```
duembgen.shape(X, init = NULL, steps = Inf, eps = 1e-06,
    maxiter = 100, in.R = FALSE, na.action = na.fail, ...)
```

**Arguments**

- `X` numeric data matrix or dataframe.
- `init` an optional matrix giving the starting value for the iteration. Otherwise the regular covariance is used after transforming it to a shape matrix with determinant 1.
- `steps` a fixed number of iteration steps to take. See details.
- `eps` convergence tolerance.
- `maxiter` maximum number of iterations.
- `in.R` logical. If TRUE R-code (and not C) is used in the iteration
- `na.action` a function which indicates what should happen when the data contain 'NA's. Default is to fail.
- `...` other arguments passed on to `tyler.shape`.

**Details**

Dümbgen’s shape matrix can be seen as `tyler.shape`’s matrix wrt to the origin for the pairwise differences of the observations. Therefore this shape matrix needs no location parameter.

The function is, however, slow if the dataset is large.

The algorithm also allows for a k-step version where the iteration is run for a fixed number of steps instead of until convergence. If `steps` is finite that number of steps is taken and `maxiter` is ignored.

**Value**

A matrix.

**Author(s)**

Klaus Nordhausen, <klaus.nordhausen@uta.fi>, Seija Sirkiä, <seija.sirkia@iki.fi>, and some of the C++ is based on work by Jari Miettinen
References


See Also

`tyler.shape`, `duembgen.shape.wt`

Examples

```r
set.seed(654321)
cov.matrix <- matrix(c(3,2,1,2,4,-0.5,1,-0.5,2), ncol=3)
X <- rmvnorm(100, c(0,0,0), cov.matrix)
cov.matrix/det(cov.matrix)^((1/3))
duembgen.shape(X)
rm(.Random.seed)
```

---

duembgen.shape.wt Weighted Duembgen's Shape Matrix

Description

Iterative algorithm to estimate the weighted version of Dümbgen’s shape matrix.

Usage

```r
duembgen.shape.wt(X, wt = rep(1, nrow(X)), init = NULL, 
eps = 1e-06, maxiter = 100, na.action = na.fail)
```

Arguments

- `X` numeric data frame or matrix.
- `wt` vector of weights. Should be nonnegative and at least one larger than zero.
- `init` an optional matrix giving the starting value for the iteration.
- `eps` convergence tolerance.
- `maxiter` maximum number of iterations.
- `na.action` a function which indicates what should happen when the data contain 'NA's. Default is to fail.

Details

The weighted Dümbgen shape matrix can be seen as `tyler.shape`’s matrix wrt to the origin for the weighted pairwise differences of the observations. Therefore this shape matrix needs no location parameter.

Note that this function is memory consuming and slow for large data sets since the matrix is based on all pairwise difference of the observations.
**hl.loc**

**Value**

a matrix.

**Author(s)**

Klaus Nordhausen, <klaus.nordhausen@uta.fi>

**References**


**See Also**

duembgen.shape

**Examples**

```r
set.seed(1)
cov.matrix.1 <- matrix(c(3,2,1,2,4,-0.5,1,-0.5,2), ncol = 3)
X.1 <- rmvnorm(100, c(0,0,0), cov.matrix.1)
cov.matrix.2 <- diag(1,3)
X.2 <- rmvnorm(50, c(1,1,1), cov.matrix.2)
X <- rbind(X.1, X.2)

D1 <- duembgen.shape.wt(X, rep(c(0,1), c(100,50)))
D2 <- duembgen.shape.wt(X, rep(c(1,0), c(100,50)))

D1
D2
```

rm(.Random.seed)

---

**hl.loc**

*Hodges - Lehmann Estimator of Location*

**Description**

Function to compute the Hodges - Lehmann estimator of location in the one sample case.

**Usage**

`hl.loc(x, na.action = na.fail)`

**Arguments**

- `x` : a numeric vector.
- `na.action` : a function which indicates what should happen when the data contain 'NA's. Default is to fail.
Details

The Hodges-Lehmann estimator is the median of the combined data points and Walsh averages. It is the same as the Pseudo Median returned as a by-product of the function `wilcox.test`.

Value

the Hodges-Lehmann estimator of location.

Author(s)

Klaus Nordhausen, <klaus.nordhausen@uta.fi>

References


See Also

`wilcox.test`

Examples

```r
set.seed(1)
x <- rt(100, df = 3)
hl.loc(x)
# same as
wilcox.test(x, conf.int = TRUE)$estimate
rm(.Random.seed)
```

---

HotellingsT2

**Hotelling’s T2 Test**

Description

Hotelling’s T2 test for the one and two sample case.

Usage

```r
HotellingsT2(X, ...)
```

## Default S3 method:
`HotellingsT2(X, Y = NULL, mu = NULL, test = "f",
na.action = na.fail, ...)

## S3 method for class 'formula'
`HotellingsT2(formula, na.action = na.fail, ...)"
**Arguments**

- **X**  
  a numeric data frame or matrix.
- **Y**  
  an optional numeric data frame or matrix for the two sample test. If NULL a one sample test is performed.
- **mu**  
  a vector indicating the hypothesized value of the mean (or difference in means if a two sample test is performed). NULL represents origin or no difference between the groups.
- **test**  
  if 'f', the decision is based on the F-distribution, if 'chi' a chi-squared approximation is used.
- **formula**  
  a formula of the form \( x \sim g \) where \( x \) is a numeric matrix giving the data values and \( g \) a factor with two levels giving the corresponding groups.
- **na.action**  
  a function which indicates what should happen when the data contain 'NA's. Default is to fail.
- **...**  
  further arguments to be passed to or from methods.

**Details**

The classical test for testing the location of a multivariate population or for testing the mean difference for two multivariate populations. When \( \text{test} = \text{"f"} \) the F-distribution is used for the test statistic and it is assumed that the data are normally distributed. If the chi-square approximation is used, the normal assumption can be relaxed to existence of second moments. In the two sample case both populations are assumed to have the same covariance matrix.

The formula interface is only applicable for the 2-sample tests.

**Value**

A list with class 'htest' containing the following components:

- **statistic**  
  the value of the T2-statistic. (That is the scaled value of the statistic that has an F distribution or a chi-square distribution depending on the value of \( \text{test} \)).
- **parameter**  
  the degrees of freedom for the T2-statistic.
- **p.value**  
  the p-value for the test.
- **null.value**  
  the specified hypothesized value of the mean or mean difference depending on whether it was a one-sample test or a two-sample test.
- **alternative**  
  a character string with the value 'two.sided'.
- **method**  
  a character string indicating what type of test was performed.
- **data.name**  
  a character string giving the name of the data (and grouping vector).

**Author(s)**

Klaus Nordhausen, <klaus.nordhausen@uta.fi>

**References**

Examples

# one sample test:

data(pulmonary)

HotellingsT2(pulmonary)
HotellingsT2(pulmonary, mu = c(0,0,2), test = "chi")

# two sample test:

set.seed(123456)
X <- rmvnorm(20, c(0, 0, 0, 0), diag(1:4))
Y <- rmvnorm(30, c(0.5, 0.5, 0.5, 0.5), diag(1:4))
Z <- rbind(X, Y)
g <- factor(rep(c(1,2),c(20,30)))

HotellingsT2(X, Y)
HotellingsT2(Z - g, mu = rep(-0.5,4))

rm(.Random.seed)

---

**HP.loc.test**

*Hallin and Paindaveine Signed-Rank Tests*

**Description**

This function implements the signed-rank location tests as suggested by Hallin and Paindaveine (2002a, 2002b).

**Usage**

```r
HP.loc.test(X, mu = NULL, score = "rank", angles = "tyler",
  method = "approximation", n.perm = 1000,
  na.action = na.fail)
```

**Arguments**

- **X**
  - a numeric data frame or matrix.
- **mu**
  - a vector indicating the hypothesized value of the location. NULL represents the origin.
- **score**
  - score for the pseudo mahalanobis distance. Options are 'rank', 'sign' and 'normal' scores.
- **angles**
  - which angle to use. Possible are 'tyler' for spatial sign type angles or 'interdirections'. Note however that currently only 'tyler' is implemented.
- **method**
  - defines the method used for the computation of the p-value. The possibilities are 'approximation' or 'permutation'.

n.perm                        if method="permutation" specifies this the number of replications used in the permutation procedure.

na.action                    a function which indicates what should happen when the data contain 'NA's. Default is to fail.

Details

The test based on interdirections is described in Hallin and Paindaveine (2002a) and the test based on Tyler's angles is described in Hallin and Paindaveine (2002b). The two different tests are asymptotically equivalent and in both cases is assumed that the data comes from an elliptic distribution.

Value

A list with class 'htest' containing the following components:

- statistic the value of the Q-statistic.
- parameter the degrees of freedom for the Q-statistic.
- p.value the p-value for the test.
- null.value the specified hypothesized value of the location.
- alternative a character string with the value 'two.sided'.
- method a character string indicating what type of test was performed.
- data.name a character string giving the name of the data.

Author(s)

Klaus Nordhausen, <klaus.nordhausen@uta.fi>

References


See Also

- tyler.shape
- spatial.sign

Examples

```r
X <- rmvnorm(100, c(0,0,0.1))
HP.loc.test(X)
HP.loc.test(X, score="s")
HP.loc.test(X, score="n")
```
HP1.shape

One Step Rank Scatter Estimator

Description

one step M-estimator of the scatter matrix based on ranks.

Usage

HP1.shape(X, location = "Estimate", na.action = na.fail, ...)

Arguments

X a numeric data frame or matrix.

location if 'Estimate' the location and scatter matrix used for computing the spatial signs are estimated simultaneously using HR.Mest, if 'Origin' or numeric tyler.shape is used with respect to origin or the given value, respectively, to obtain the spatial signs.

na.action a function which indicates what should happen when the data contain 'NA's. Default is to fail.

... arguments that can be passed on to tyler.shape or HR.Mest.

Details

This is a one step M-estimator of shape which is standardized in such a way that the determinant is 1.

The exact formula is:

\[ V = V_0^{\frac{1}{2}} \text{ave}\left\{ a\left(\frac{R_i}{n+1}\right)u_i'u_i\right\} V_0^{\frac{1}{2}}. \]

where \( V_0 \) is Tyler’s shape matrix, \( u_i = ||z_i||^{-1}z_i \) is the spatial sign of \( z_i = (x_i - \mu) V_0^{-\frac{1}{2}} \) and \( R_i \) gives the rank of \( ||z|| \) among \( ||z_1||, \ldots, ||z_n|| \). The van der Warden score function \( a(.) \) is the inverse of the cdf of a chi-squared distribution with \( p \) degrees of freedom.

This scatter matrix is based on the test for shape developed in the paper by Hallin and Paindaveine (2006), its usage with respect to the origin is demonstrated in Nordhausen et al. (2006).

Author(s)

Klaus Nordhausen, <klaus.nordhausen@uta.fi>

References


Examples

```r
set.seed(654321)
cov.matrix <- matrix(c(3,2,1,2,4,-0.5,1,-0.5,2), ncol=3)
X <- rmvnorm(100, c(0,0,0), cov.matrix)
HP1.shape(X)
HP1.shape(X, location="Origin")
cov.matrix/det(cov.matrix)^(1/3)
rm(.Random.seed)
```

---

**HR.Mest**

*Simultaneous Affine Equivariant Estimation of Multivariate Median and Tyler’s Shape Matrix*

---

**Description**

iterative algorithm that finds the affine equivariant multivariate median by estimating `tyler.shape` simultaneously.

**Usage**

```r
HR.Mest(X, maxiter = 100, eps.scale = 1e-06, eps.center = 1e-06,
        na.action = na.fail)
```

**Arguments**

- `X` a numeric data frame or matrix.
- `maxiter` maximum number of iterations.
- `eps.scale` convergence tolerance for the Tyler’s shape matrix subroutine.
- `eps.center` convergence tolerance for the location estimate.
- `na.action` a function which indicates what should happen when the data contain `NA`'s. Default is to fail.

**Details**

The algorithm follows the idea of Hettmansperger and Randles (2002). There are, however, some differences. This algorithm has the vector of marginal medians as starting point for the location and the starting shape matrix is Tyler’s shape matrix based on the vector of marginal medians and has then a location step and a shape step which are:

**location step k+1:** transforming the data as \( y = x V_k^{-\frac{1}{2}} \) and computing the spatial median \( \mu_y \) of \( y \) using the function `spatial.median`. Then retransforming \( \mu_y \) to the original scale \( \mu_{x,k+1} = \mu_y V_k^{\frac{1}{2}} \).

**shape step k+1:** computing Tyler’s shape matrix \( V_{k+1} \) with respect to \( \mu_{x,k+1} \) by using the function `tyler.shape`.
The algorithm stops when the difference between two subsequent location estimates is smaller than \( \text{eps.center} \).
There is no proof that the algorithm converges.

**Value**

A list containing:
- **center** vector with the estimated location.
- **scatter** matrix of the estimated scatter.

**Author(s)**

Klaus Nordhausen, <klaus.nordhausen@uta.fi>, and Seija Sirkiä, <seija.sirkia@iki.fi>

**References**


**Examples**

```r
set.seed(654321)
cov.matrix <- matrix(c(3,2,1,2,4,-0.5,1,-0.5,2), ncol=3)
X <- rmvnorm(100, c(0,0,0), cov.matrix)
res <- HR.Mest(X)
colMeans(X)
res$center
cov.matrix/det(cov.matrix)^(1/3)
resscatter
rm(.Random.seed)
```

---

**ind.ctest**

*Test of Independence based on Marginal Ranks*

**Description**

Performs the test that a group of variables is independent of another based on marginal ranks. Three different score functions are available.

**Usage**

```r
ind.ctest(X, index1, index2 = NULL, scores = "rank",
 na.action = na.fail)
```
Arguments

X: a data frame or matrix.

index1: integer vector that selects the columns of X that form group one. Only numeric columns can be selected.

index2: integer vector that selects the columns of X that form group two. Only numeric columns can be selected. If NULL, all remaining columns of X will be selected.

scores: if 'sign', a sign test is performed, if 'rank' a rank test is performed or if 'normal' a normal score test is performed.

na.action: a function which indicates what should happen when the data contain 'NA's. Default is to fail.

Details

The test tests if X[, index1] is independent of X[, index2] and is described in great detail in Puri and Sen (1971).

Value

A list with class 'htest' containing the following components:

- statistic: the value of the W-statistic.
- parameter: the degrees of freedom for the W-statistic.
- p.value: the p-value for the test.
- method: a character string indicating what type of test was performed.
- data.name: a character string giving the name of the data.

Author(s)

Klaus Nordhausen, <klaus.nordhausen@uta.fi>

References


Examples

A1 <- matrix(c(4, 4, 5, 4, 6, 6, 5, 6, 7), ncol = 3)
A2 <- matrix(c(0.5, -0.3, -0.3, 0.7), ncol = 2)
X <- cbind(rmvnorm(100, c(-1, 0, 1), A1), rmvnorm(100, c(0, 0, 0), A2))
ind.ctest(X, 1:3)
ind.ctest(X, c(1, 5), c(2, 3), scores = "normal")
**ind.ictest**

*Test of Independence based on Marginal Ranks in a Symmetric IC Model*

**Description**

Performs the test that a group of variables is independent of an other based on marginal ranks. It is assumed that the data follows a symmetric IC model. Three different score functions are available.

**Usage**

```r
ind.ictest(X, index1, index2 = NULL, scores = "rank",
method = "approximation", n.simu = 1000,
..., na.action = na.fail)
```

**Arguments**

- `X` a data frame or matrix.
- `index1` integer vector that selects the columns of `X` that form group one. Only numeric columns can be selected.
- `index2` integer vector that selects the columns of `X` that form group two. Only numeric columns can be selected. If `NULL`, all remaining columns of `X` will be selected.
- `scores` if `"sign"`, a sign test is performed, if `"rank"` a signed rank test is performed or if `"normal"` a normal score test is performed.
- `method` defines the method used for the computation of the p-value. The possibilites are "approximation" (default), "simulation" or "permutation". Details below.
- `n.simu` if `method = "simulation"` or `method = "permutation"` this specifies the number of replications used in the simulation or permutation procedure.
- `...` further arguments to be passed to the function `ics`
- `na.action` a function which indicates what should happen when the data contain `NA's`. Default is to fail.

**Details**

Assumed is here that `X[, index1]` comes from a symmetric independent component model which in turn is independent from `X[, index2]` which has also an underlying symmetric independent component model. This function recovers the independent components using the function `ics`, centers them by a marginal location estimate based on the same scores that will be used in the actual test. The test is described in Oja, Paindaveine and Taskinen (2009). The asymptotic chi-square distribution is however even for large sample sizes inadequat and therefore p-values can be simulated by resampling the test statistic under the null hypothesis or by permuting the rows of the independent components of `X[, index2]`. Both alternatives are also described in Oja, Paindaveine and Taskinen (2009).
Value

A list with class 'htest' containing the following components:

- statistic: the value of the Q-statistic.
- parameter: the degrees of freedom for the Q-statistic or the number of replications depending on the chosen method.
- p.value: the p-value for the test.
- method: a character string indicating what type of test was performed.
- data.name: a character string giving the name of the data.

Author(s)

Klaus Nordhausen, <klaus.nordhausen@uta.fi>

References


Examples

```r
Z1 <- cbind(rt(500,5), rnorm(500), runif(500))
Z2 <- cbind(rt(500,8), rbeta(500,2,2))
A1 <- matrix(c(4, 4, 5, 4, 6, 6, 5, 6, 7), ncol = 3)
A2 <- matrix(c(0.5, -0.3, -0.3, 0.7), ncol = 2)
X <- cbind(Z1 %*% t(A1), Z2 %*% t(A2))

ind.ictest(X, 1:3)
ind.ictest(X, 1:3, method = "simu")
ind.ictest(X, 1:2, 3:5, method = "perm", S1 = tyler.shape, S2 = cov)
```

---

**LASERI**

*Cardiovascular Responses to Head-up Tilt*

Description

This data set contains the cardiovascular responses to a passive head-up tilt for 223 subjects.

Usage

```r
data(LASERI)
```
### Format

A data frame with 223 observations on the following 32 variables.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sex</td>
<td>a factor with levels Female and Male.</td>
</tr>
<tr>
<td>age</td>
<td>Age in years.</td>
</tr>
<tr>
<td>height</td>
<td>Height in cm.</td>
</tr>
<tr>
<td>weight</td>
<td>Weight in kg.</td>
</tr>
<tr>
<td>waist</td>
<td>Waist circumference in cm.</td>
</tr>
<tr>
<td>hip</td>
<td>Hip circumference in cm.</td>
</tr>
<tr>
<td>bmi</td>
<td>Body mass index.</td>
</tr>
<tr>
<td>whr</td>
<td>Waist hip ratio.</td>
</tr>
<tr>
<td>hrt1</td>
<td>Average heart rate in the tenth minute of rest.</td>
</tr>
<tr>
<td>hrt2</td>
<td>Average heart rate in the second minute during the tilt.</td>
</tr>
<tr>
<td>hrt3</td>
<td>Average heart rate in the fifth minute during the tilt.</td>
</tr>
<tr>
<td>hrt4</td>
<td>Average heart rate in the fifth minute after the tilt.</td>
</tr>
<tr>
<td>cot1</td>
<td>Average cardiac output in the tenth minute of rest.</td>
</tr>
<tr>
<td>cot2</td>
<td>Average cardiac output in the second minute during the tilt.</td>
</tr>
<tr>
<td>cot3</td>
<td>Average cardiac output in the fifth minute during the tilt.</td>
</tr>
<tr>
<td>cot4</td>
<td>Average cardiac output in the fifth minute after the tilt.</td>
</tr>
<tr>
<td>svrit1</td>
<td>Average systemic vascular resistance index in the tenth minute of rest.</td>
</tr>
<tr>
<td>svrit2</td>
<td>Average systemic vascular resistance index in the second minute during the tilt.</td>
</tr>
<tr>
<td>svrit3</td>
<td>Average systemic vascular resistance index in the fifth minute during the tilt.</td>
</tr>
<tr>
<td>svrit4</td>
<td>Average systemic vascular resistance index in the fifth minute after the tilt.</td>
</tr>
<tr>
<td>pwvt1</td>
<td>Average pulse wave velocity in the tenth minute of rest.</td>
</tr>
<tr>
<td>pwvt2</td>
<td>Average pulse wave velocity in the second minute during the tilt.</td>
</tr>
<tr>
<td>pwvt3</td>
<td>Average pulse wave velocity in the fifth minute during the tilt.</td>
</tr>
<tr>
<td>pwvt4</td>
<td>Average pulse wave velocity in the fifth minute after the tilt.</td>
</tr>
<tr>
<td>hrt1t2</td>
<td>Difference hrt1 - hrt2.</td>
</tr>
<tr>
<td>cot1t2</td>
<td>Difference cot1 - cot2.</td>
</tr>
<tr>
<td>svrit1t2</td>
<td>Difference svrit1 - svrit2.</td>
</tr>
<tr>
<td>pwvt1t2</td>
<td>Difference pwvt1 - pwvt2.</td>
</tr>
<tr>
<td>hrt1t4</td>
<td>Difference hrt1 - hrt4.</td>
</tr>
<tr>
<td>cot1t4</td>
<td>Difference cot1 - cot4.</td>
</tr>
<tr>
<td>svrit1t4</td>
<td>Difference svrit1 - svrit4.</td>
</tr>
<tr>
<td>pwvt1t4</td>
<td>Difference pwvt1 - pwvt4.</td>
</tr>
</tbody>
</table>
Details
This data is a subset of hemodynamic data collected as a part of the LASERI study (English title: “Cardiovascular risk in young Finns study”) using whole-body impedance cardiography and plethysmographic blood pressure recordings from fingers. The data given here comes from 223 healthy subjects between 26 and 42 years of age, who participated in the recording of the hemodynamic variables both in a supine position and during a passive head-up tilt on a motorized table. During that experiment the subject spent the first ten minutes in a supine position, then the motorized table was tilted to a head-up position (60 degrees) for five minutes, and for the last five minutes the table was again returned to the supine position.

Of interest in this data is for example if the values 5 minutes after the tilt are already returned to their pre-tilt levels.

Source
Data courtesy of the LASERI study (http://youngfinnsstudy.utu.fi/).

Examples
# for example testing if the location before the tilt is the same as
# 5 minutes after the tilt:
data(LASERI)
DIFFS.T1T4 <- subset(LASERI, select=c(HRT1T4,C01T4,SVR1T4))
rankctest(DIFFS.T1T4)
rankctest(DIFFS.T1T4, score="s")

pair.diff

**Pairwise Differences**

Description
Computes pairwise differences.

Usage
pair.diff(X)

Arguments
X a numeric matrix.

Details
The function computes all differences of row i and row j with i < j. The function is a wrapper to a C function to do the computation quickly and does no checks concerning the input.
Pairwise Products

Description
Computes pairwise elementwise products.

Usage
   pair.prod(X)

Arguments
   X     a numeric matrix.

Details
The function computes all elementwise products of row i and row j with i < j. The function is a
wrapper to a C function to do the computation quickly and does no checks concerning the input.

Value
Matrix containing the products.

Author(s)
Klaus Nordhausen, <klaus.nordhausen@uta.fi>

See Also
   pair.diff, pair.sum

Examples
   X <- matrix(1:10, ncol = 2, byrow = FALSE)
   pair.diff(X)
Examples

X <- matrix(1:10, ncol = 2, byrow = FALSE)
pair.prod(X)

Description

Computes pairwise sums.

Usage

pair.sum(X)

Arguments

X        a numeric matrix.

Details

The function computes all sums of row i and row j with i < j. The function is a wrapper to a C function to do the computation quickly and does no checks concerning the input.

Value

Matrix containing the sums.

Author(s)

Seija Sirkiä, <seija.sirkia@iki.fi>

See Also

pair.diff, pair.prod

Examples

X <- matrix(1:10, ncol = 2, byrow = FALSE)
pair.sum(X)
Change in Pulmonary Response after Exposure to Cotton Dust

Description

Changes in pulmonary function of 12 workers after 6 hours of exposure to cotton dust.

Usage

data(pulmonary)

Format

A data frame with 12 observations on the following 3 variables.

FVC  change in FVC (forced vital capacity) after 6 hours.
FEV  change in FEV\_3 (forced expiratory volume) after 6 hours.
CC   change in CC (closing capacity) after 6 hours.

Note

There is also a different version of this data set around. In the different version the FVC value of subject 11 is -0.01 instead of -0.10.

Source


References


Examples

data(pulmonary)
plot(pulmonary)
Description

Performs the one, two or c sample location test based on marginal ranks. Three different score functions are available.

Usage

\texttt{rank.ctest(X, \ldots)}

## Default S3 method:
\texttt{rank.ctest(X, Y = NULL, mu = NULL, scores = "rank",}
\texttt{ na.action = na.fail, \ldots)}

## S3 method for class 'formula'
\texttt{rank.ctest(formula, na.action = na.fail, \ldots)}

## S3 method for class 'ics'
\texttt{rank.ctest(X, g = NULL, index = NULL, na.action = na.fail, \ldots)}

Arguments

\begin{itemize}
  \item \texttt{X} \quad a numeric data frame or matrix or an ics object.
  \item \texttt{Y} \quad an optional numeric data frame or matrix for the two sample test. If NULL a one sample test is performed.
  \item \texttt{mu} \quad a vector indicating the hypothesized value of the mean (or difference in means if you are performing a two sample test). NULL represents origin or no difference between the groups. For more than two groups \texttt{mu} should be 0 or not be specified at all.
  \item \texttt{scores} \quad if 'sign', a sign test is performed, if 'rank' a signed rank test is performed or if 'normal' a normal score test is performed.
  \item \texttt{formula} \quad a formula of the form \texttt{X \sim g} where \texttt{X} is a numeric matrix giving the data values and \texttt{g} a factor with at least two levels giving the corresponding groups.
  \item \texttt{g} \quad a grouping factor with at least two levels.
  \item \texttt{index} \quad an integer vector that gives the columns to choose the invariant coordinates form the 'ics' object. The default uses all columns.
  \item \texttt{na.action} \quad a function which indicates what should happen when the data contain 'NA's. Default is to fail.
  \item \texttt{\ldots} \quad further arguments to be passed to or from methods.
\end{itemize}
Details

These tests are well described in Puri and Sen (1971). The tests are based on the marginal ranks for which three score functions are available. The scores are also used to estimate the covariance matrices. In the multisample case it is assumed that the distribution of the different populations differs only in their location.

The ics interface provides an invariant test based on the invariant coordinate selection. The asymptotic distribution is however still an open question when more than one component is used, though the chi-square approximation works well also for several components as shown in Nordhausen, Oja and Tyler (2006).

Value

A list with class 'htest' containing the following components:

- statistic: the value of the T-statistic.
- parameter: the degrees of freedom for the T-statistic.
- p.value: the p-value for the test.
- null.value: the specified hypothesized value of the mean or mean difference depending on whether it was a one-sample test or a two-sample test.
- alternative: a character string with the value 'two.sided'.
- method: a character string indicating what type of test was performed.
- data.name: a character string giving the name of the data (and grouping vector).

Author(s)

Klaus Nordhausen, <klaus.nordhausen@uta.fi>

References


Examples

```r
# one sample tests:

data(pulmonary)

rank.ctest(pulmonary, scores = "sign")
rank.ctest(pulmonary, mu = c(0,0,2))

# two sample tests:

set.seed(123456)
X <- rmvnorm(20, c(0,0,0), diag(1:4))
Y <- rmvnorm(30, c(0.5,0.5,0.5,0.5), diag(1:4))
```
**rank.ictest**

One Sample Location Test based on Marginal Ranks in the Independent Component Model

**Description**

marginal rank test for the location problem in the one sample case when the margins are assumed independent.

**Usage**

```r
rank.ictest(X, ...)  
## Default S3 method:  
rank.ictest(X, mu = NULL, scores = "rank", method = "approximation",  
            n.simu = 1000, na.action = na.fail, ...)  
## S3 method for class 'ics'  
rank.ictest(X, index = NULL, na.action = na.fail, ...)  
```

```r
Z <- rbind(X,Y)  
g <- factor(rep(c(1,2), c(20,30)))  

rank.ictest(X, Y, scores = "normal")  
rank.ictest(Z-g, scores = "sign", mu = rep(-0.5,4))  

# c sample test:  
W <- rmvnorm(30, c(0,0,0,0), diag(1:4))  
Z2 <- rbind(X,Y,W)  
g2 <- factor(rep(1:3, c(20,30,30)))  

rank.ictest(Z2-g2, scores = "normal")  

# in an invariant coordinate system  
rank.ictest(ics(Z2,covOrigin, cov4, S2args=list(location = "Origin"), index = c(1,4), scores = "sign")  
rank.ictest(ics(Z), g, index = 4)  
rank.ictest(ics(Z2), g2, scores = "normal",index = 4)  
rm(.Random.seed)
```
Arguments

Arguments

- **X**
  - a numeric data frame or matrix or an ics object.

- **mu**
  - a vector indicating the hypothesized value of the location. NULL represents the origin.

- **scores**
  - options are 'rank' for the signed rank test, 'sign' for the sign test and 'normal' for the normal score test.

- **method**
  - defines the method used for the computation of the p-value. The possibilities are "approximation" (default), "simulation" or "permutation". Details below.

- **n.simu**
  - if 'method=simulation' or 'method=permutation' this specifies the number of replications used in the simulation or permutation procedure.

- **index**
  - an integer vector that gives the columns to choose from invariant coordinates form the 'ics' object. The default uses all columns.

- **na.action**
  - a function which indicates what should happen when the data contain 'NA's. Default is to fail.

- **...**
  - further arguments to be passed to or from methods.

Details

The test is normally used to test for location in the symmetric independent component model.

By default the limiting distribution is used to compute the p-values. However for moderate sample sizes (N=50) was observed in Nordhausen et al. (2009) that the normal score test can be sometimes slightly biased. Therefore the argument method can be used to get p-values based on simulations from a multivariate normal under the null or by permuting the signs of the centered observations.

Value

A list with class 'htest' containing the following components:

- **statistic**
  - the value of the Q-statistic.

- **parameter**
  - the degrees of freedom for the Q-statistic.

- **p.value**
  - the p-value for the test.

- **null.value**
  - the specified hypothesized value of the location.

- **alternative**
  - a character string with the value 'two.sided'.

- **method**
  - a character string indicating what type of test was performed.

- **data.name**
  - a character string giving the name of the data.

Author(s)

Klaus Nordhausen, <klaus.nordhausen@uta.fi>

References

Examples

```r
set.seed(555)
X <- cbind(rt(30, 8), rnorm(30, 0.5), runif(30, -3, 3))
mix.matrix <- matrix(c(3, 2, 1, 2, 4, -0.5, 1, -0.5, 2), ncol=3)
X.mixed <- X %*% t(mix.matrix)
ica.X <- ics(X, covOrigin, cov4, S2args = list(location = "Origin"))
r ank.ictest(ica.X)
r ank.ictest(ica.X, scores = "normal", method = "simu")
r ank.ictest(ics.components(ica.X), scores = "normal", method = "perm")
rm(.Random.seed)
```

---

**spatial.median**

*Spatial Median*

**Description**

iterative algorithm to compute the spatial median.

**Usage**

```r
spatial.median(X, init = NULL, maxiter = 500, eps = 1e-06,
                print.it = FALSE, na.action = na.fail)
```

**Arguments**

- **X**: a numeric data frame or data matrix.
- **init**: Starting value for the algorithm, if 'NULL', the vector of marginal medians is used.
- **maxiter**: maximum number of iterations.
- **eps**: convergence tolerance.
- **print.it**: logical. If TRUE prints the number of iterations, otherwise not.
- **na.action**: a function which indicates what should happen when the data contain 'NA's. Default is to fail.

**Details**

Follows the algorithm of Vardi and Zhang.

**Value**

vector of the spatial median.

**Author(s)**

Klaus Nordhausen, <klaus.nordhausen@uta.fi> and Seija Sirkiä, <seija.sirkia@iki.fi>
References


Examples

```r
set.seed(654321)
cov.matrix <- matrix(c(3,2,1,2,4,-0.5,1,-0.5,2), ncol=3)
X <- rmvnorm(100, c(0,0,0), cov.matrix)
spatial.median(X)
rm(.Random.seed)
```

spatial.sign Spatial Signs

Description

Function to obtain the spatial signs of a multivariate dataset. The function can compute the spatial signs also with respect to a given or estimated loacation and scale. If both location and scale have to be estimated the hrNmest function is used, if only one has to be estimated the, estimation is done using spatial.median or tyler.shape.

Usage

```r
spatial.sign(X, center = TRUE, shape = TRUE, 
               na.action = na.fail, ...)
```

Arguments

- `X` a numeric data frame or matrix.
- `center` either a logical value or a numeric vector of length equal to the number of columns of ’X’. See below for more information.
- `shape` either a logical value or a square numeric matrix with number of columns equal to the number of columns of ’X’. See below for more information.
- `na.action` a function which indicates what should happen when the data contain ’NA’s. Default is to fail.
- `...` arguments that can be passed on to functions used for the estimation of location and shape.
Details

The spatial signs U of X with location $\mu$ and shape V are given by

$$u_i = \frac{(x_i - \mu)V^{-\frac{1}{2}}}{||(x_i - \mu)V^{-\frac{1}{2}}||}.$$ 

If a numeric value is given as 'center' and/or 'shape' these are used as $\mu$ and/or V in the above formula. If 'center' and/or 'shape' are 'TRUE' the values for $\mu$ and/or V are estimated, if 'FALSE' the origin is used as the value of $\mu$ and/or the identity matrix as the value of V.

In the special case of univariate data the univariate signs of the data (centered if requested) are returned and the shape parameter is redundant.

Value

a matrix with the spatial signs of the data as rows or the univariate signs as a px1 matrix. The centering vector and scaling matrix used are returned as attributes 'center' and 'shape'.

Author(s)

Klaus Nordhausen, <klaus.nordhausen@uta.fi> and Seija Sirkiä, <seija.sirkia@iki.fi>

See Also

HR.Mest

Examples

```r
set.seed(654321)
cov.matrix <- matrix(c(3,2,1,2,4,-0.5,1,-0.5,2), ncol=3)
X <- rmvnorm(15, c(1,0,-1), cov.matrix)
spatial.sign(X)
spatial.sign(X, center=FALSE, shape=FALSE)
spatial.sign(X, center=colMeans(X), shape=cov(X))
rm(.Random.seed)
```

---

**symm.huber**

*Symmetrized Huber Scatter Matrix*

Description

Iterative algorithm to estimate the symmetrized Huber scatter matrix.

Usage

```r
symm.huber(X, qg = 0.9, init = NULL, eps = 1e-06, maxiter = 100,
na.action = na.fail)
```
Arguments

- **x**: numeric data frame or matrix.
- **qg**: tuning parameter. Should be between 0 and 1. The default is 0.9.
- **init**: an optional matrix giving the starting value for the iteration.
- **eps**: convergence tolerance.
- **maxiter**: maximum number of iterations.
- **na.action**: a function which indicates what should happen when the data contain 'NA's. Default is to fail.

Details

The symmetrized Huber scatter matrix is the regular Huber scatter matrix for the pairwise differences of the observations taken wrt to the origin.

Note that this function might be memory consuming and slow for large data sets since the matrix is based on all pairwise difference of the observations.

The function `symmhuber` in the package **SpatialNP** offers also a k-step option. The **SpatialNP** package contains also the function `mvhuberM` for the regular multivariate Huber location and scatter estimator.

Value

a matrix.

Author(s)

Klaus Nordhausen, <klaus.nordhausen@uta.fi>, and Jari Miettinen

References


See Also

`symm.huber.wt`, `symmhuber`, `mvhuberM`

Examples

```r
set.seed(654321)
cov.matrix <- matrix(c(3,2,1,2,4,-0.5,1,-0.5,2), ncol=3)
X <- rmvnorm(100, c(0,0,0), cov.matrix)
symm.huber(X)
rm(.Random.seed)
```
symm.huber.wt

Weighted Symmetrized Huber Scatter Matrix

Description
Iterative algorithm to estimate the weighted symmetrized Huber scatter matrix.

Usage
```r
symm.huber.wt(X, wt = rep(1, nrow(X)), qg = 0.9, init = NULL,
              eps = 1e-06, maxiter = 100, na.action = na.fail)
```

Arguments
- `X`: numeric data frame or matrix.
- `wt`: vector of weights. Should be nonnegative and at least one larger than zero.
- `qg`: tuning parameter. Should be between 0 and 1. The default is 0.9.
- `init`: an optional matrix giving the starting value for the iteration.
- `eps`: convergence tolerance.
- `maxiter`: maximum number of iterations.
- `na.action`: a function which indicates what should happen when the data contain `NA’s. Default is to fail.

Details
The weighted symmetrized Huber scatter matrix is the regular Huber scatter matrix for the weighted pairwise differences of the observations taken wrt to the origin.

Note that this function is memory consuming and slow for large data sets since the matrix is based on all pairwise difference of the observations.

Value
a matrix.

Author(s)
Klaus Nordhausen, <klaus.nordhausen@uta.fi>

References

See Also
symm.huber
tyler.shape

**Tyler’s Shape Matrix**

**Description**

Iterative algorithm to estimate Tyler’s shape matrix.

**Usage**

```r
tyler.shape(X, location = NULL, init = NULL, steps = Inf, eps = 1e-06, 
maxiter = 100, in.R = FALSE, print.it = FALSE, 
na.action = na.fail)
```

**Arguments**

- `X` numeric data matrix or dataframe.
- `location` if NULL the sample mean is used, otherwise a vector with the location can be specified.
- `init` an optional matrix giving the starting value for the iteration
- `steps` a fixed number of iteration steps to take. See details.
- `eps` convergence tolerance.
- `maxiter` maximum number of iterations.
- `in.R` logical. If TRUE R-code (and not C) is used in the iteration
- `print.it` logical. If TRUE prints the number of iterations, otherwise not.
- `na.action` a function which indicates what should happen when the data contain 'NA's. Default is to fail.

**Examples**

```r
set.seed(1)
cov.matrix.1 <- matrix(c(3,2,1,2,4,-0.5,1,-0.5,2), ncol = 3)
X.1 <- rmvnorm(100, c(0,0,0), cov.matrix.1)
cov.matrix.2 <- diag(1,3)
X.2 <- rmvnorm(50, c(1,1,1), cov.matrix.2)
X <- rbind(X.1, X.2)

d1 <- symm.huber.wt(X, rep(c(0,1), c(100,50)))
d2 <- symm.huber.wt(X, rep(c(1,0), c(100,50)))

D1
d2

rm(.Random.seed)
```
Details

The most robust M-estimator of shape. It is proportional to the regular covariance matrix for elliptical contoured distributions. The estimate is in such a way standardized, that its determinate is 1.

The algorithm requires an estimate of location, if none is provided, the sample mean is used. Observations which are equal to the location estimate are removed from the data.

The algorithm also allows for a k-step version where the iteration is run for a fixed number of steps instead of until convergence. If steps is finite that number of steps is taken and maxiter is ignored.

Value

A matrix.

Author(s)

Klaus Nordhausen, <klaus.nordhausen@uta.fi>, and Seija Sirkiä, <seija.sirkia@iki.fi>

References


See Also

duembgen.shape, HR.Mest

Examples

```r
set.seed(654321)
cov.matrix <- matrix(c(3,2,1,2,4,-0.5,1,-0.5,2), ncol=3)
X <- rmvnorm(100, c(0,0,0), cov.matrix)
tyler.shape(X)
tyler.shape(X, location=0)
cov.matrix/det(cov.matrix)^{(1/3)}
rm(.Random.seed)
```

## vdw.loc

Van der Waerden Estimator of Location

Description

Iterative algorithm to compute the location estimator based on van der Waerden scores (sometimes also referred to as normal scores).

Usage

```
vdw.loc(x, int.diff = 10, maxiter = 1000, na.action = na.fail)
```
Arguments

- **x**: a numeric vector.
- **int.diff**: number of observations in internal interval when the estimate is searched.
- **maxiter**: maximum number of iterations.
- **na.action**: a function which indicates what should happen when the data contain 'NA's. Default is to fail.

Details

The algorithm searches among the observations and all Walsh averages for the two points nearest around the root of the van der Waerden score criterion. Since the criterion function is monotone first the int.diff of the sorted data points are searched that contain the root. After then determining there the two points of question a linear interpolation is used as an estimate.

Value

the van der Waerden score estimator of location.

Author(s)

Klaus Nordhausen, <klaus.nordhausen@uta.fi>

References


Examples

```r
set.seed(1)
x <- rt(100, df = 3)
vdw.loc(x)
rm(.Random.seed)
```
Index

*Topic datasets
LASERI, 15
pulmonary, 20

*Topic htest
HotellingsT2, 6
HP.loc.test, 8
ind.ctest, 12
ind.ictest, 14
rank.ctest, 21
rank.ictest, 23
spatial.median, 25
spatial.sign, 26
vdw.loc, 31

*Topic package
ICSNP-package, 2

*Topic robust
duembgen.shape, 3
duembgen.shape.wt, 4
hl.loc, 5
symm.huber, 27
symm.huber.wt, 29
tyler.shape, 30
vdw.loc, 31

*Topic univar
hl.loc, 5

duembgen.shape, 3, 5, 31
duembgen.shape.wt, 4, 4

dl.loc, 5
HotellingsT2, 6
HP.loc.test, 8
HP1.shape, 10
HR.Mest, 10, 11, 26, 27, 31
ICS, 2
ics, 14
ICSNP (ICSNP-package), 2
ICSNP-package, 2
ind.ctest, 12
ind.ictest, 14

LASERI, 15

pair.diff, 17, 18, 19
pair.prod, 18, 19
pair.sum, 18, 19
pulmonary, 20
rank.c.test, 21
rank.i.test, 23

spatial.median, 25, 26
spatial.sign, 9, 26
symm.huber, 27, 29
symm.huber.wt, 28, 29

tyler.shape, 3, 4, 9–11, 26, 30

vdw.loc, 31

wilcox.test, 6