Package 'SpatialNP'

July 21, 2025

Type Package

Title Multivariate Nonparametric Methods Based on Spatial Signs and Ranks

Version 1.1-6

Date 2025-03-27

Author Seija Sirkia [aut],
Jari Miettinen [aut] (ORCID: <https://orcid.org/0000-0002-3270-7014>),
Klaus Nordhausen [cre, aut] (ORCID:
<https://orcid.org/0000-0002-3758-8501>),
Hannu Oja [aut] (ORCID: <https://orcid.org/0000-0002-4945-5976>),
Sara Taskinen [aut] (ORCID: <https://orcid.org/0000-0001-9470-7258>)

Maintainer Klaus Nordhausen <klausnordhausenr@gmail.com>

Depends R (>= 2.4.0)

Suggests ICSNP, MNM

Description Test and estimates of location, tests of independence, tests of sphericity and several estimates of shape all based on spatial signs, symmetrized signs, ranks and signed ranks. For details, see Oja and Randles (2004) <doi:10.1214/088342304000000558> and Oja (2010) <doi:10.1007/978-1-4419-0468-3>.

License GPL-2

NeedsCompilation yes

Repository CRAN

Date/Publication 2025-03-27 18:30:02 UTC

Contents

`SpatialNP-package` *Multivariate Nonparametric Methods Based on Spatial Signs and Ranks*

Description

Test and estimates of location, tests of independence, tests of sphericity and several estimates of shape all based on spatial signs, symmetrized signs, ranks and signed ranks. For details, see Oja and Randles (2004) <doi:10.1214/088342304000000558> and Oja (2010) <doi:10.1007/978-1-4419-0468-3>.

Details

Package:	SpatialNP
Type:	Package
Title:	Multivariate Nonparametric Methods Based on Spatial Signs and Ranks
Version:	1.1-6
Date:	2025-03-27
Authors@R:	c(person(given = "Seija", family = "Sirkia", role = "aut"), person(given = "Jari", family = "Miettinen", role = "a
Author:	Seija Sirkia [aut], Jari Miettinen [aut] (<https://orcid.org/0000-0002-3270-7014>), Klaus Nordhausen [cre, aut
Maintainer:	Klaus Nordhausen <klausnordhausenr@gmail.com>
Depends:	R (>= 2.4.0)
Suggests:	ICSNP, MNM
Description:	Test and estimates of location, tests of independence, tests of sphericity and several estimates of shape all base
License:	GPL-2
Archs:	x64

There are three functions for inference, `sr.loc.test`, `sr.indep.test` and `sr.sphere.test`, for location, independence and sphericity tests. The so called inner and outer standardization matrices are also available as well as the actual sign and rank score functions, together with a utility function `to.shape`.

Author(s)

Seija Sirkia [aut], Jari Miettinen [aut] (<https://orcid.org/0000-0002-3270-7014>), Klaus Nordhausen [cre, aut] (<https://orcid.org/0000-0002-3758-8501>), Hannu Oja [aut] (<https://orcid.org/0000-0002-4945-5976>), Sara Taskinen [aut] (<https://orcid.org/0000-0001-9470-7258>)

Maintainer: Klaus Nordhausen <klausnordhausenr@gmail.com>

See Also

package **ICSNP**
package **MNM**

`Independence tests` *Multivariate test of independence based on spatial signs or ranks*

Description

Test of independence between two sets of variables. Inference is based on the spatial signs of the observations, symmetrized signs of the observations or spatial signed ranks of the observations.

Usage

```
 sr.indep.test(X, Y = NULL, g = NULL, score = c("sign",
"symmsign", "rank"), regexp = FALSE, cond = FALSE, cond.n = 1000,
na.action = na.fail)
```

Arguments

`X`	a matrix or a data frame
`Y`	an optional matrix or a data frame
`g`	a factor giving the two sets of variables, or numeric vector or vector of column names giving the first set of variables. See details
`score`	a character string indicating which transformation of the observations should be used
`regexp`	logical. Is g a regular expression?
`cond`	logical. Should the conditionally distribution free test be used?
`cond.n`	Number of permutations to use in the conditionally distribution free test
`na.action`	a function which indicates what should happen when the data contain 'NA's. Default is to fail.

Details

`X` should contain the first set of variables and `Y` the second with matching rows. Alternatively, `X` should contain both sets and `g` should be a factor of length equal to number of columns of `X`, or, `g` should be a numeric or character vector naming the variables in the first set. If `g` is a character vector it is assumed to name all wanted columns exactly, unless `regexp` is `TRUE`.

Value

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

`statistic`	the value of the statistic
`parameter`	the degrees of freedom for the statistic or the number of replications if conditionally distribution free p-value was used
`p.value`	the p-value for the test
`null.value`	the specified hypothesized value of the measure of dependence (always 0)
`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)

Seija Sirkia, <seija.sirkia@iki.fi>

References

Taskinen, S., Oja, H., Randles R. (2004) Multivariate Nonparametric Tests of Independence. JASA, *100, 916-925*

See Also

Spatial signs and ranks

Examples

```
A<-matrix(c(1,2,-3,4,3,-2,-1,0,4),ncol=3)
X<-matrix(rnorm(3000),ncol=3)%*%t(A)
Y<-cbind(X+runif(3000,-1,1),runif(1000))
sr.indep.test(X,Y)
#alternative calls:
Z<-cbind(X,Y)
colnames(Z)<-c("a1","a2","a3","b1","b2","b3","b4")
g<-factor(c(rep(1,3),rep(2,4)))
sr.indep.test(Z,g=g)
sr.indep.test(Z,g=c("b"),regexp=TRUE)
sr.indep.test(Z,g=1:3)
```

`Location tests` *Spatial sign and rank tests of multivariate location*

Description

Multivariate tests of location of one or more samples based on spatial signs and (signed) ranks. In case of one sample the null hypothesis about a given location is tested. In case of several samples the null hypothesis is that all samples have the same location.

Usage

```
 sr.loc.test(X, Y = NULL, g = NULL, score = c("sign", "rank"),
nullvalue = NULL, cond = FALSE, cond.n = 1000,
na.action = na.fail,...)
```

Arguments

`X`	a matrix or a data frame
`Y`	an optional matrix or a data frame
`g`	a factor giving the groups (may contain just one level)
`score`	a character string indicating which transformation of the observations should be used
`nullvalue`	location to be tested in the one sample case (ignored if there is more than one sample)
`cond`	logical. Should the conditionally distribution free test be used? (Ignored if `score` is `"rank"`)
`cond.n`	number of permutations to use in the conditionally distribution free test
`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 other functions

Details

`X` should contain the the whole data set and `g` should describe the groups, or, if there is only one group, `g` may be missing. Alternatively, if there are two samples `X` may contain only the first sample while the second sample is given in `Y` and `g` is ignored. Note that in the one sample case when `rank` is chosen as `score` the function in fact uses signed ranks.

Note that the conditionally distribution free p-value is only provided for the sign based version of the test.

Value

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

statistic	the value of the statistic
parameter	the degrees of freedom for the statistic or the number of replications if conditionally distribution free p-value was used
p.value	the p-value for the test
null.value	the specified hypothesized value of the (common) 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 (and grouping vector)

Author(s)

Seija Sirkia, <seija.sirkia@iki.fi>

References

Oja, H., Randles R. (2004) Multivariate Nonparametric Tests. Statistical Science *19, 598-605.*

See Also

Spatial signs and ranks

Examples

```
A<-matrix(c(1,2,-3,4,3,-2,-1,0,4),ncol=3)
X<-rbind(matrix(rnorm(1500),ncol=3),matrix(rnorm(750)+1,ncol=3))%*%t(A)
sr.loc.test(X,cond=TRUE)
X[1:250,]<-X[1:250,]+1
g<-factor(rep(c(1,2,3),each=250))
sr.loc.test(X,g=g,score="rank")
```

---

mvhuberM *Multivariate Huber's M-estimator and its symmetrized version*

Description

Iterative algorithms to estimate M-estimators of location and scatter as well as symmetrized M-estimator using Huber's weight functions.

Usage

```
mvhuberM(X, qg = 0.9, fixed.loc = FALSE, location = NULL, init =
NULL, steps = Inf, eps = 1e-06, maxiter = 100, na.action = na.fail)

symmhuber(X, qg = 0.9, init = NULL, steps = Inf, eps = 1e-6,
maxiter = 100, na.action = na.fail)

symmhuber.inc(X, qg=0.9, m=10, init=NULL, steps=Inf, permute=TRUE,
eps=1e-6, maxiter=100, na.action = na.fail)
```

Arguments

X	a matrix or a data frame
qg	a tuning parameter. The default is 0.9, see details
fixed.loc	a logical, see details
location	an optional vector giving the location of the data or the initial value for the location if it is estimated
init	an optional starting value for scatter
steps	fixed number of iteration steps to take, if Inf iteration is repeated until convergence (or until maxiter steps)
m	a parameter in symmhuber.inc which defines how many pairwise differences are used, see details.
permute	logical in symmhuber.inc which determines whether the rows of X are permuted randomly, see details.
eps	tolerance for convergence
maxiter	maximum number of iteration steps. Ignored if steps is finite
na.action	a function which indicates what should happen when the data contain 'NA's. Default is to fail.

Details

mvhuberM computes multivariate M-estimators of location and scatter using Huber's weight functions. The tuning parameter qg defines cutoff-point c for weight functions so that $c = F^{-1}(q)$, where F is the cdf of χ^2-distribution with p degrees of freedom. The estimators with maximal breakdown point are obtained with the choice qg=F(p+1). If fixed.loc is set TRUE, scatter estimator is computed with fixed location given by location (default is column means).

symmhuber computes Huber's M-estimator of scatter using pairwise differences of the data therefore avoiding location estimation.

symmhuber.inc is a computationally lighter estimator to approximate symmetrized Huber's M-estimator of scatter. Only a subset of the pairwise differences are used in the computation in the incomplete case. The magnitude of the subset used is controlled by the argument m which is half of the number of how many differences each observation is part of. Differences of successive observations are used, and therefore random permutation of the rows of X is suggested and is the default choice in the function. For details see Miettinen et al., 2016.

Value

`mvhuberM` returns a list with components

`location`	a vector
`scatter`	a matrix

`symmhuber` returns a matrix.

`symmhuber.inc` returns a matrix.

Author(s)

Klaus Nordhausen, <klaus.nordhausen@tuwien.ac.at>,
Jari Miettinen, <jari.p.miettinen@aalto.fi>

References

Huber, P.J. (1981), Robust Statistics, Wiley, New York.

Lopuhaa, H.P. (1989). On the relation between S-estimators and M-estimators of multivariate location and covariance. Annals of Statistics, *17, 1662-1683.*

Sirkia, S., Taskinen, S., Oja, H. (2007) Symmetrised M-estimators of scatter. Journal of Multivariate Analysis, *98, 1611-1629.*

Miettinen, J., Nordhausen, K., Taskinen, S., Tyler, D.E. (2016) On the computation of symmetrized M-estimators of scatter. In Agostinelli, C. Basu, A., Filzmoser, P. and Mukherje, D. (editors) "Recent Advances in Robust Statistics: Theory and Application", *131-149, Springer India, New Delhi.*

Examples

```
A<-matrix(c(1,2,-3,4,3,-2,-1,0,4),ncol=3)
X<-matrix(rnorm(1500),ncol=3)%*%t(A)
mvhuberM(X)
symmhuber(X)
symmhuber.inc(X, m=5)
symm.mvtmle.inc(X, m=5)
```

Shape matrices *Shape matrices based on spatial ranks and signed ranks*

Description

Iterative algorithms to find shape matrices based on spatial signs and ranks and the k-step versions of these.

Usage

```
spatial.shape(X, score = c("sign", "symmsign", "rank", "signrank"),
fixed.loc = FALSE, location = NULL, init = NULL, steps = Inf,
eps = 1e-06, maxiter = 100, na.action = na.fail)

signs.shape(X, fixed.loc = FALSE, location = NULL, init = NULL,
steps = Inf, eps = 1e-6, maxiter = 100, na.action = na.fail)

symmsign.shape(X, init = NULL, steps = Inf, eps = 1e-6,
maxiter = 100, na.action = na.fail)

symmsign.shape.inc(X, m=10, init=NULL, steps=Inf, permute=TRUE,
eps=1e-6, maxiter=100, na.action=na.fail)

rank.shape(X, init = NULL, steps = Inf, eps = 1e-06,
maxiter = 100, na.action = na.fail)

signrank.shape(X, fixed.loc = FALSE, location = NULL, init = NULL,
steps = Inf, eps = 1e-06, maxiter = 100, na.action = na.fail)
```

Arguments

`X`	a matrix or a data frame
`score`	a character string indicating which transformation of the observations should be used
`fixed.loc`	a logical, see details
`location`	an optional vector giving the location of the data or the initial value for the location if it is estimated
`init`	an optional starting value for the iteration
`steps`	fixed number of iteration steps to take, if `Inf` iteration is repeated until convergence (or until `maxiter` steps)
`m`	a parameter in `symmsign.shape.inc` which defines how many pairwise differences are used, see details
`permute`	logical in `symmsign.shape.inc` which determines whether the rows of `X` are permuted randomly, see details.
`eps`	tolerance for convergence
`maxiter`	maximum number of iteration steps. Ignored if `steps` is finite
`na.action`	a function which indicates what should happen when the data contain 'NA's. Default is to fail.

Details

`sign.shape` is Tyler's shape matrix and `symmsign.shape` is Duembgen's shape matrix. Function `symmsign.shape.inc` is for a computationally lighter estimator to approximate Duembgen's shape

matrix. Only a subset of the pairwise differences are used in the computation in the incomplete case. The magnitude of the subset used is controlled by the argument `m` which is half of the number of how many differences each observation is part of. Differences of successive observations are used, and therefore random permutation of the rows of `X` is suggested and is the default choice in the function. For details see Miettinen et al., 2016. `rank.shape` and `signrank.shape` are the so called inner standardization matrices of location etc. tests based on spatial signs and ranks. When data is standardized using these matrices the corresponding sign or rank scores will appear "uncorrelated": the corresponding outer standardization matrices will be proportional to the identity matrix, see examples.

`spatial.shape` is a wrapper function for a unified access to all four shape estimates (not including `symmsign.shape.inc`). The choice of estimate is done via `score`:

- `"sign"` for `signs.shape`
- `"symmsign"` for `symmsign.shape`
- `"rank"` for `rank.shape`
- `"signrank"` for `signrank.shape`

`signrank.shape` and `sign.shape` include options to compute the shape matrix either with respect to fixed location (`fixed.loc = TRUE`) or so that the location and the shape are estimated simultaneously (`fixed.loc = FALSE`).

Value

The estimate matrix with the (final estimate of or given) location vector as attribute `"location"`.

Author(s)

Seija Sirkia, <seija.sirkia@iki.fi>, Jari Miettinen, <jari.p.miettinen@aalto.fi>

References

Oja, H., Randles R. (2004) Multivariate Nonparametric Tests. Statistical Science *19, 598-605.*

Sirkia et al. (2009) Tests and estimates of shape based on spatial signs and ranks. Journal of Nonparametric Statistics, *21, 155-176.*

Sirkia, S., Taskinen, S., Oja, H. (2007) Symmetrised M-estimators of scatter. Journal of Multivariate Analysis, *98, 1611-1629.*

Miettinen, J., Nordhausen, K., Taskinen, S., Tyler, D.E. (2016) On the computation of symmetrized M-estimators of scatter. In Agostinelli, C. Basu, A., Filzmoser, P. and Mukherje, D. (editors) "Recent Advances in Robust Statistics: Theory and Application", *131-149, Springer India, New Delhi.*

See Also

`tyler.shape`, `duembgen.shape`, also spatial sign and rank covariance matrices and spatial signs and ranks

Examples

```
A<-matrix(c(1,2,-3,4,3,-2,-1,0,4),ncol=3)
X<-matrix(rnorm(1500),ncol=3)%*%t(A)
symmsign.shape(X)
to.shape(symmsign.shape(X),trace=3)
spatial.shape(X,score="sign")
spatial.shape(X,score="sign",fixed.loc=TRUE)
to.shape(A%*%t(A))
# one-step shape estimate based on spatial ranks and covariance matrix:
spatial.shape(X,score="rank",init=cov(X),steps=1)
symmsign.shape.inc(X, m=5)
```

`Spatial location` *Multivariate location estimates based on spatial signs and signed ranks*

Description

Iterative algorithms to find spatial median, multivariate Hodges-Lehmann estimate of location, their affine equivariant versions and k-step versions of these.

Usage

```
spatial.location(X, score = c("sign", "signrank"), init = NULL,
shape = TRUE, steps = Inf, maxiter = 500, eps = 1e-6,
na.action = na.fail)

ae.spatial.median(X, init = NULL, shape = TRUE, steps = Inf,
maxiter = 500, eps = 1e-6, na.action = na.fail)

ae.hl.estimate(X, init = NULL, shape = TRUE, steps = Inf,
maxiter = 500, eps = 1e-06, na.action = na.fail)
```

Arguments

`X`	a matrix or a data frame
`score`	a character string indicating which transformation of the observations should be used
`init`	an optional vector giving the initial point of the iteration
`shape`	logical, or a matrix. See details
`steps`	fixed number of iteration steps to take, if `Inf` iteration is repeated until convergence (or until `maxiter` steps)
`eps`	tolerance for convergence
`maxiter`	maximum number of iteration steps
`na.action`	a function which indicates what should happen when the data contain 'NA's. Default is to fail.

Details

Spatial median and Hodges-Lehmann estimator (spatial median of the pairwise differences) are not affine equivariant. Affine equivariance can be achieved by simultaneously estimating the corresponding shape, as proposed for the spatial median by Hettmansperger and Randles (2002). For spatial median the corresponding shape is `signs.shape` and for the Hodges-Lehmann estimate it is `signrank.shape`.

`spatial.location` is a wrapper function for a unified access to both location estimates. The choice of estimate is done via `score`:

- `"sign"` for spatial median
- `"signrank"` for Hodges-Lehmann estimate

If a matrix (must be symmetric and positive definite, but this is not checked) is given as `shape` the location estimate is found with respect to that shape and no further shape estimation is done. If a logical `TRUE` is given as `shape` the shape is estimated and consequently the affine equivariant version of the location estimate is found. If `shape` is `FALSE` then shape estimation is not done and the non affine equivariant versions of the location estimate, that is the spatial median and the Hodges-Lehmann estimate are found.

Value

The estimate vector with the (final estimate of or given) shape matrix as attribute `"shape"`.

Author(s)

Seija Sirkia, <seija.sirkia@iki.fi>, Jari Miettinen, <jari.p.miettinen@aalto.fi>

References

Hettmansperger, T. and Randles, R. (2002) A Practical Affine Equivariant Multivariate Median, Biometrika, 89, pp. 851-860

See Also

`spatial.median`, `signrank.shape`

Examples

```
A<-matrix(c(1,2,-3,4,3,-2,-1,0,4),ncol=3)
X<-matrix(rnorm(3000),ncol=3)%*%t(A)
spatial.location(X,score="signrank")
spatial.location(X,score="sign")
#compare with:
colMeans(X)
ae.hl.estimate(X,shape=A%*%t(A))
ae.hl.estimate(X,shape=FALSE)
```

`Spatial sign and rank covariance matrices`
Spatial sign and rank covariance matrices

Description

Functions to compute spatial sign, spatial symmetrized sign, spatial rank and spatial signed rank covariance matrices

Usage

```
SCov(X, location = NULL, na.action = na.fail)
SSCov(X, na.action = na.fail)
RCov(X, na.action = na.fail)
SRCov(X, location = NULL, na.action = na.fail)
```

Arguments

`X`	matrix or a data frame
`location`	numeric vector (may be missing)
`na.action`	a function which indicates what should happen when the data contain 'NA's. Default is to fail.

Details

These functions compute the matrices of the form

$$ave\{S(x_i)S^T(x_i)\}$$

where $S(x_i)$ are the appropriate scores of the data: spatial signs, spatial symmetrized signs, spatial ranks or spatial signed ranks. These are the so called outer standardization matrices of location etc. tests based on spatial signs and ranks. They are not affine equivariant.

`SCov` and `SRCov` require a location vector with respect to which they are computed. If none is provided, `SCov` uses spatial median and `SRCov` uses Hodges-Lehmann estimator.

Author(s)

Seija Sirkia, `<seija.sirkia@iki.fi>`

References

Visuri, S., Koivunen, V. and Oja, H. (2000). Sign and rank covariance matrices. J. Statistical Planning and Inference, *91, 557-575.*

See Also

spatial signs and ranks, corresponding shape matrices (inner standardization matrices)

Examples

```
A<-matrix(c(1,2,-3,4,3,-2,-1,0,4),ncol=3)
X<-matrix(rt(150,1),ncol=3)%*%t(A)
SCov(X)
SSCov(X)
RCov(X)
SRCov(X)
to.shape(A%*%t(A),trace=1)
```

Spatial signs, symmetrized signs, ranks and signed ranks

Spatial signs, symmetrized signs, ranks and signed ranks

Description

Functions to compute spatial signs, symmetrized signs, ranks and signed ranks.

Usage

```
spatial.signs(X, center = TRUE, shape = TRUE,
na.action = na.fail,...)
spatial.symmsign(X, shape = TRUE, na.action = na.fail, ...)
spatial.rank(X, shape = TRUE, na.action = na.fail, ...)
spatial.signrank(X, center = TRUE, shape = TRUE,
na.action = na.fail,...)
```

Arguments

X	a matrix or a data frame
center	a vector or a logical, see details
shape	a matrix or a logical, see details
...	arguments that can be passed on to function used for the estimation of shape.
na.action	a function which indicates what should happen when the data contain 'NA's. Default is to fail.

Details

The spatial signs of an observed vector is simply the vector, possibly affinely transformed first, multiplied by its Euclidian length. See spatial.sign for a precise definition. Symmetrized spatial signs are the spatial signs of the pairwise differences of the data

$$||x_i - x_j||^{-1}(x_i - x_j)$$

(there are n over 2 of these). Spatial rank of an observation is the average of the signs of the differences of that observation and the others:

$$R(x_i) = ave_j\{||x_i - x_j||^{-1}(x_i - x_j)\}$$

Spatial signed rank of an observation is defined as

$$Q(x_i) = (R(x_i) + ave_j\{||x_i + x_j||^{-1}(x_i + x_j)\})/2$$

If a numerical value is given for `shape` and/or `center` these are used to transform the data before the computation of signs or ranks. A logical `TRUE` indicates that the shape or center should be estimated. In this case an affine transformation that makes the resulting signs or ranks have a covariance matrix equal or proportional to the identity matrix and centerd on the origin is found. A logical `FALSE` indicates that the null value, that is, the identity matrix or the origin, should be used. Note that only signed ranks depend on a center.

The value of shape and/or location used are returned as attributes.

Author(s)

Seija Sirkia, <seija.sirkia@iki.fi>

References

Visuri, S., Koivunen, V. and Oja, H. (2000). Sign and rank covariance matrices. J. Statistical Planning and Inference, *91*, *557-575.*

See Also

`spatial.sign` for the signs, spatial sign and rank covariance matrices and `spatial.shape` for the standardizing transformations

Examples

```
A<-matrix(c(1,2,-3,4),ncol=2)
X<-matrix(rnorm(100),ncol=2)%*%t(A)
def.par<-par(no.readonly=TRUE) # for resetting
layout(matrix(1:4,ncol=2,nrow=2,byrow=TRUE))
plot(X,col=c(2,rep(1,19)))
plot(spatial.symmsign(X),col=c(2,rep(1,19)),xlim=c(-1,1),ylim=c(-1,1))
theta<-seq(0,2*pi,length=1000)
lines(sin(theta),cos(theta))
plot(spatial.rank(X),col=c(2,rep(1,19)),xlim=c(-1,1),ylim=c(-1,1))
lines(sin(theta),cos(theta))
plot(spatial.signrank(X),col=c(2,rep(1,19)),xlim=c(-1,1),ylim=c(-1,1))
lines(sin(theta),cos(theta))
par(def.par)
```

`Sphericity tests` *Sphericity tests*

Description

Tests of sphericity based on spatial signs and spatial signs of pairwise differences.

Usage

```
 sr.sphere.test(X, score = c("sign", "symmsign"), shape = NULL,
na.action = na.fail)
```

Arguments

X	a matrix or a data frame
score	a character string indicating which transformation of the observations should be used
shape	a matrix with which the data should be standardized before the sphericity test
na.action	a function which indicates what should happen when the data contain 'NA's. Default is to fail.

Details

The test is for a null hypothesis of the form "true shape matrix is equal to the identity matrix". Effectively, giving a matrix as shape will produce a test of whether the true shape is equal (in fact, proportional, since the scale of shape will have no effect) to it. In that case the test will still be for sphericity but the data is standardized beforehand.

Value

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

statistic	the value of the statistic
parameter	the degrees of freedom for the statistic
p.value	the p-value for the test
null.value	the specified hypothesized value of the shape (always diag(p) where p is the number of dimensions)
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)

Seija Sirkia, <seija.sirkia@iki.fi>

References

Sirkia et al. (2009) Tests and estimates of shape based on spatial signs and ranks. Journal of Nonparametric Statistics, *21, 155-176.*

See Also

sign and rank covariance matrices

Examples

```
A<-matrix(c(1,2,-3,4,3,-2,-1,0,4),ncol=3)
X<-matrix(rnorm(600),ncol=3)%*%t(A)
sr.sphere.test(X,score="sign")
```

`symm.mvtmle` *Symmetrized M-estimators of scatter with the weights of the t-distribution*

Description

Iterative algorithms to estimate symmetrized M-estimators of scatter using weights of the t-distribution.

Usage

```
symm.mvtmle(X, nu=1, init=NULL, steps=Inf, eps=1e-6,
maxiter=100, na.action = na.fail)

symm.mvtmle.inc(X, nu=1, m=10, init=NULL, steps=Inf, permute=TRUE,
eps=1e-6, maxiter=100, na.action = na.fail)
```

Arguments

`X`	a matrix or a data frame
`nu`	the degrees of freedom of the t-distribution. The default is 1. Must be larger than 0.
`init`	an optional starting value for scatter
`steps`	fixed number of iteration steps to take, if `Inf` iteration is repeated until convergence (or until `maxiter` steps)
`m`	a parameter in `symm.mvtmle.inc` which defines how many pairwise differences are used, see details.
`permute`	logical in `symm.mvtmle.inc` which determines whether the rows of `X` are permuted randomly, see details.
`eps`	tolerance for convergence
`maxiter`	maximum number of iteration steps. Ignored if `steps` is finite
`na.action`	a function which indicates what should happen when the data contain 'NA's. Default is to fail.

Details

`symm.mvtmle` computes M-estimator of scatter using weights of the t-distribution and pairwise differences of the data. Hence, location estimation is not needed.

`symm.mvtmle.inc` is a computationally lighter estimator to approximate symmetrized M-estimator of scatter which uses weights of the t-distribution. Only a subset of the pairwise differences are

used in the computation in the incomplete case. The magnitude of the subset used is controlled by the argument `m` which is half of the number of how many differences each observation is part of. Differences of successive observations are used, and therefore random permutation of the rows of `X` is suggested and is the default choice in the function. For details see Miettinen et al., 2016.

Value

`symm.mvtmle` returns a matrix.

`symm.mvtmle.inc` returns a matrix.

Author(s)

Jari Miettinen, <jari.p.miettinen@aalto.fi>,
Klaus Nordhausen, <klaus.nordhausen@tuwien.ac.at>

References

Huber, P.J. (1981), Robust Statistics, Wiley, New York.

Sirkia, S., Taskinen, S., Oja, H. (2007) Symmetrised M-estimators of scatter. Journal of Multivariate Analysis, *98, 1611-1629.*

Duembgen, L., Pauly, M., Schweizer, T. (2015) M-Functionals of multivariate scatter. Statistics Surveys 9, *32-105.*

Miettinen, J., Nordhausen, K., Taskinen, S., Tyler, D.E. (2016) On the computation of symmetrized M-estimators of scatter. In Agostinelli, C. Basu, A., Filzmoser, P. and Mukherje, D. (editors) "Recent Advances in Robust Statistics: Theory and Application", *131-149, Springer India, New Delhi.*

Examples

```
A<-matrix(c(1,2,-3,4,3,-2,-1,0,4),ncol=3)
X<-matrix(rnorm(1500),ncol=3)%*%t(A)
symm.mvtmle(X, nu=2)
symm.mvtmle.inc(X, nu=2, m=20)
```

---

to.shape *Rescale a matrix to a shape matrix*

Description

This function rescales a given matrix such that its determinant, trace or the value of the first diagonal element meets a criteria.

Usage

```
to.shape(M, determ, trace, first)
```

Arguments

M	a matrix to be scaled
determ	required value for determinant
trace	required value for trace
first	required value of the first diagonal element

Details

If determ, trace or first is given M is scaled such that its determinant, trace or first diagonal element, respectively, equals that value. If none of the three is given M is scaled such that its determinant equals one. If more than one criteria is given the first of them is used and the others silently ignored.

Value

The rescaled matrix

Note

A shape matrix is symmetric and positive definite square matrix. In order for the result to be such the argument matrix M should also be symmetric and positive definite square matrix. However, the function does not check for this. Expect to see errors if M is of inappropriate type.

Author(s)

Seija Sirkia, <seija.sirkia@iki.fi>

References

Paindaveine D. (2008) A Canonical Definition of Shape. Statistics and Probability Letters 78, 2240-2247

Examples

```
(A<-matrix(c(1,2,-3,4,3,-2,-1,0,4),ncol=3))
to.shape(A%*%t(A))
to.shape(A%*%t(A),trace=3)
to.shape(A%*%t(A),first=1)
```

Index