

Package ‘energy’

April 17, 2009

Title E-statistics (energy statistics)

Version 1.1-0

Date 2008-04-07

Author Maria L. Rizzo and Gabor J. Szekely

Description E-statistics (energy) tests and statistics for comparing distributions: multivariate normality, multivariate k-sample test for equal distributions, hierarchical clustering by e-distances, multivariate independence tests, distance correlation, goodness-of-fit tests. Energy-statistics concept based on a generalization of Newton’s potential energy is due to Gabor J. Szekely.

Maintainer Maria Rizzo <mrizzo@bgnet.bgsu.edu>

Depends boot

License GPL (>= 2)

Repository CRAN

Date/Publication 2008-04-07 17:48:05

R topics documented:

dcov.test	2
distance correlation	4
edist	6
energy.hclust	8
eqdist.etest	10
indep.etest	12
indep.test	13
ksample.e	16
mvI	17
mvI.test	18
mvnorm.etest	19
poisson.mtest	21

Index	23
--------------	-----------

dcov.test

Distance Covariance Test

Description

Distance covariance test of multivariate independence. Distance covariance and distance correlation are multivariate measures of dependence.

Usage

```
dcov.test(x, y, index = 1.0, R = 199)
```

Arguments

x	matrix: first sample, observations in rows
y	matrix: second sample, observations in rows
R	number of replicates
index	exponent on Euclidean distance, in (0,2]

Details

dcov.test performs a nonparametric test of multivariate independence. The test decision is obtained via bootstrap, with R replicates.

The sample sizes (number of rows) of the two samples must agree, and samples must not contain missing values. The statistic is $n\mathcal{V}_n^2$ where $\mathcal{V}_n(x, y) = \text{dcov}(x, y)$, which is based on interpoint Euclidean distances $\|x_i - y_j\|$.

Distance correlation is a new measure of dependence between random vectors introduced by Szekely, Rizzo, and Bakirov (2007). For all distributions with finite first moments, distance correlation \mathcal{R} generalizes the idea of correlation in two fundamental ways: (1) $\mathcal{R}(X, Y)$ is defined for X and Y in arbitrary dimension. (2) $\mathcal{R}(X, Y) = 0$ characterizes independence of X and Y .

Distance correlation satisfies $0 \leq \mathcal{R} \leq 1$, and $\mathcal{R} = 0$ only if X and Y are independent. Distance covariance \mathcal{V} provides a new approach to the problem of testing the joint independence of random vectors. The formal definitions of the population coefficients \mathcal{V} and \mathcal{R} are given in (SRB 2007). The definitions of the empirical coefficients are given in the energy [dcov](#) topic.

For all values of the index in (0,2) (all except 2), the asymptotic distribution of $n\mathcal{V}_n^2$ is a quadratic form of centered Gaussian random variables, with coefficients that depend on the distributions of X and Y . For the general problem of testing independence when the distributions of X and Y are unknown, the test based on $n\mathcal{V}_n^2$ can be implemented as a permutation test. See (SRB 2007) for theoretical properties of the test, including statistical consistency.

Value

dcov.test returns a list with class `htest` containing

method	description of test
--------	---------------------

statistic	observed value of the test statistic
estimate	dCov(x,y)
estimates	a vector: [dCov(x,y), dCor(x,y), dVar(x), dVar(y)]
replicates	replicates of the test statistic
p.value	approximate p-value of the test
data.name	description of data

Note

For the test of independence, the distance covariance test statistic is the V-statistic $n \text{dCov}^2 = n \mathcal{V}_n^2$ (not dCov).

Author(s)

Maria L. Rizzo (mrizzo@bgnnet.bgsu.edu) and Gabor J. Szekely (gabors@bgnnet.bgsu.edu)

References

Szekely, G.J., Rizzo, M.L., and Bakirov, N.K. (2007), Measuring and Testing Dependence by Correlation of Distances, *Annals of Statistics*, Vol. 35 No. 6, pp. 2769-2794.
<http://dx.doi.org/10.1214/009053607000000505>

See Also

[dcov](#) [dcor](#) [DCOR](#)

Examples

```
## independent multivariate data
x <- matrix(rnorm(60), nrow=20, ncol=3)
y <- matrix(rnorm(40), nrow=20, ncol=2)
dcov.test(x, y, R = 99)

## Not run:
## dependent multivariate data
library(MASS)
Sigma <- matrix(c(1, .1, 0, 0, 1, 0, 0, .1, 1), 3, 3)
x <- mvrnorm(30, c(0, 0, 0), .1 * diag(3))
y <- mvrnorm(30, c(0, 0, 0), Sigma) * x
set.seed(123); dcov.test(x, y, index = 1.5)
set.seed(123); dcov.test(x, y)
detach("package:MASS")

## End(Not run)
```

 distance correlation

Distance Correlation and Covariance Statistics

Description

Computes distance covariance and distance correlation statistics, which are multivariate measures of dependence.

Usage

```
dcov(x, y, index = 1.0)
dcor(x, y, index = 1.0)
DCOR(x, y, index = 1.0)
```

Arguments

x	matrix: first sample, observations in rows
y	matrix: second sample, observations in rows
index	exponent on Euclidean distance, in (0,2]

Details

dcov and dcor or DCOR compute distance covariance and distance correlation statistics. DCOR is a self-contained R function returning a list of statistics. dcor execution is faster than DCOR (see examples).

The sample sizes (number of rows) of the two samples must agree, and samples must not contain missing values.

Distance correlation is a new measure of dependence between random vectors introduced by Szekely, Rizzo, and Bakirov (2007). For all distributions with finite first moments, distance correlation \mathcal{R} generalizes the idea of correlation in two fundamental ways: (1) $\mathcal{R}(X, Y)$ is defined for X and Y in arbitrary dimension. (2) $\mathcal{R}(X, Y) = 0$ characterizes independence of X and Y .

Distance correlation satisfies $0 \leq \mathcal{R} \leq 1$, and $\mathcal{R} = 0$ only if X and Y are independent. Distance covariance \mathcal{V} provides a new approach to the problem of testing the joint independence of random vectors. The formal definitions of the population coefficients \mathcal{V} and \mathcal{R} are given in (SRB 2007). The definitions of the empirical coefficients are as follows.

The empirical distance covariance $\mathcal{V}_n(\mathbf{X}, \mathbf{Y})$ with index 1 is the nonnegative number defined by

$$\mathcal{V}_n^2(\mathbf{X}, \mathbf{Y}) = \frac{1}{n^2} \sum_{k,l=1}^n A_{kl} B_{kl}$$

where A_{kl} and B_{kl} are

$$A_{kl} = a_{kl} - \bar{a}_{k.} - \bar{a}_{.l} + \bar{a}_{..}$$

$$B_{kl} = b_{kl} - \bar{b}_{k.} - \bar{b}_{.l} + \bar{b}_{..}$$

Here

$$a_{kl} = \|X_k - X_l\|_p, \quad b_{kl} = \|Y_k - Y_l\|_q, \quad k, l = 1, \dots, n,$$

and the subscript \cdot denotes that the mean is computed for the index that it replaces. Similarly, $\mathcal{V}_n(\mathbf{X})$ is the nonnegative number defined by

$$\mathcal{V}_n^2(\mathbf{X}) = \mathcal{V}_n^2(\mathbf{X}, \mathbf{X}) = \frac{1}{n^2} \sum_{k, l=1}^n A_{kl}^2.$$

The empirical distance correlation $\mathcal{R}_n(\mathbf{X}, \mathbf{Y})$ is the square root of

$$\mathcal{R}_n^2(\mathbf{X}, \mathbf{Y}) = \frac{\mathcal{V}_n^2(\mathbf{X}, \mathbf{Y})}{\sqrt{\mathcal{V}_n^2(\mathbf{X})\mathcal{V}_n^2(\mathbf{Y})}}.$$

See `dcov.test` for a test of multivariate independence based on the distance covariance statistic.

Value

`dcov` returns the sample distance covariance and `dcor` returns the sample distance correlation. `DCOR` returns a list with elements

<code>dCov</code>	sample distance covariance
<code>dCor</code>	sample distance correlation
<code>dVarX</code>	distance variance of x sample
<code>dVarY</code>	distance variance of y sample

Note

Two methods of computing the statistics are provided. `DCOR` is a stand-alone R function that returns a list of statistics. `dcov` and `dcor` provide R interfaces to the C implementation, which is usually faster. `dcov` and `dcor` call an internal function `.dcov`.

Note that it is inefficient to compute `dCor` by:

```
square root of dcov(x, y) / sqrt(dcov(x, x) * dcov(y, y))
```

because the individual calls to `dcov` involve unnecessary repetition of calculations. For this reason, both `.dcov` and `DCOR` compute and return all four statistics.

Author(s)

Maria L. Rizzo (mrizzo@bgnnet.bgsu.edu) and Gabor J. Szekely (gabors@bgnnet.bgsu.edu)

References

Szekely, G.J., Rizzo, M.L., and Bakirov, N.K. (2007), Measuring and Testing Dependence by Correlation of Distances, *Annals of Statistics*, Vol. 35 No. 6, pp. 2769-2794.

<http://dx.doi.org/10.1214/0090536070000000505>

See Also

`dcov.test`

Examples

```
## independent multivariate data
x <- matrix(rnorm(30), nrow=20, ncol=3)
y <- matrix(rnorm(40), nrow=20, ncol=2)

## C implementation
dcov(x, y, 1.5)
dcor(x, y, 1.5)
.dcov(x, y, 1.5)
## R implementation
DCOR(x, y, 1.5)

## Not run:
## compare speed of R version and C version
set.seed(111)
## R version
system.time(replicate(1000, DCOR(x, y)))
set.seed(111)
## C version
system.time(replicate(1000, .dcov(x, y)))

## End(Not run)
```

edist

E-distance

Description

Returns the E-distances (energy statistics) between clusters.

Usage

```
edist(x, sizes, distance = FALSE, ix = 1:sum(sizes), alpha = 1)
```

Arguments

x	data matrix of pooled sample or Euclidean distances
sizes	vector of sample sizes
distance	logical: if TRUE, x is a distance matrix
ix	a permutation of the row indices of x
alpha	distance exponent

Details

A vector containing the pairwise two-sample multivariate \mathcal{E} -statistics for comparing clusters or samples is returned. The e-distance between clusters is computed from the original pooled data, stacked in matrix `x` where each row is a multivariate observation, or from the distance matrix `x` of the original data, or distance object returned by `dist`. The first `sizes[1]` rows of the original data matrix are the first sample, the next `sizes[2]` rows are the second sample, etc. The permutation vector `ix` may be used to obtain e-distances corresponding to a clustering solution at a given level in the hierarchy.

The e-distance between two clusters C_i, C_j of size n_i, n_j proposed by Szekely and Rizzo (2003) is the e-distance $e(C_i, C_j)$, defined by

$$e(C_i, C_j) = \frac{n_i n_j}{n_i + n_j} [2M_{ij} - M_{ii} - M_{jj}],$$

where

$$M_{ij} = \frac{1}{n_i n_j} \sum_{p=1}^{n_i} \sum_{q=1}^{n_j} \|X_{ip} - X_{jq}\|^\alpha,$$

$\|\cdot\|$ denotes Euclidean norm, $\alpha = \text{alpha}$, and X_{ip} denotes the p -th observation in the i -th cluster. The exponent `alpha` should be in the interval $(0, 2]$.

Value

A object of class `dist` containing the lower triangle of the e-distance matrix of cluster distances corresponding to the permutation of indices `ix` is returned.

Author(s)

Maria L. Rizzo `<mrizzo @ bgnet.bgsu.edu>` and Gabor J. Szekely `<gabors @ bgnet.bgsu.edu>`

References

Szekely, G. J. and Rizzo, M. L. (2005) Hierarchical Clustering via Joint Between-Within Distances: Extending Ward's Minimum Variance Method, *Journal of Classification* 22(2) 151-183.

<http://dx.doi.org/10.1007/s00357-005-0012-9>

Szekely, G. J. and Rizzo, M. L. (2004) Testing for Equal Distributions in High Dimension, *InterStat*, November (5).

Szekely, G. J. (2000) Technical Report 03-05, \mathcal{E} -statistics: Energy of Statistical Samples, Department of Mathematics and Statistics, Bowling Green State University.

See Also

[energy.hclust](#) [eqdist](#) [etest](#) [ksample.e](#)

Examples

```
## compute e-distances for 3 samples of iris data
data(iris)
edist(iris[,1:4], c(50,50,50))
```

```
## compute e-distances from vector of group labels
d <- dist(matrix(rnorm(100), nrow=50))
g <- cutree(energy.hclust(d), k=4)
edist(d, sizes=table(g), ix=rank(g, ties.method="first"))
```

energy.hclust

Hierarchical Clustering by Minimum (Energy) E-distance

Description

Performs hierarchical clustering by minimum (energy) E-distance method.

Usage

```
energy.hclust(dst, alpha = 1)
```

Arguments

dst	Euclidean distances in a <code>dist</code> object, or a distance matrix produced by <code>dist</code> , or lower triangle of distance matrix as vector in column order. If <code>dst</code> is a square matrix, the lower triangle is interpreted as a vector of distances.
alpha	distance exponent

Details

Dissimilarities are $d(x, y) = \|x - y\|^\alpha$, where the exponent α is in the interval (0,2]. This function performs agglomerative hierarchical clustering. Initially, each of the n singletons is a cluster. At each of $n-1$ steps, the procedure merges the pair of clusters with minimum e-distance. The e-distance between two clusters C_i, C_j of sizes n_i, n_j is given by

$$e(C_i, C_j) = \frac{n_i n_j}{n_i + n_j} [2M_{ij} - M_{ii} - M_{jj}],$$

where

$$M_{ij} = \frac{1}{n_i n_j} \sum_{p=1}^{n_i} \sum_{q=1}^{n_j} \|X_{ip} - X_{jq}\|^\alpha,$$

$\|\cdot\|$ denotes Euclidean norm, and X_{ip} denotes the p -th observation in the i -th cluster.

The return value is an object of class `hclust`, so `hclust` methods such as `print` or `plot` methods, `plclust`, and `cutree` are available. See the documentation for `hclust`.

The e-distance measures both the heterogeneity between clusters and the homogeneity within clusters. \mathcal{E} -clustering ($\alpha = 1$) is particularly effective in high dimension, and is more effective than some standard hierarchical methods when clusters have equal means (see example below). For other advantages see the references.

Value

An object of class `hclust` which describes the tree produced by the clustering process. The object is a list with components:

`merge`: an $n-1$ by 2 matrix, where row i of `merge` describes the merging of clusters at step i of the clustering. If an element j in the row is negative, then observation $-j$ was merged at this stage. If j is positive then the merge was with the cluster formed at the (earlier) stage j of the algorithm.

`height`: the clustering height: a vector of $n-1$ non-decreasing real numbers (the e-distance between merging clusters)

`order`: a vector giving a permutation of the indices of original observations suitable for plotting, in the sense that a cluster plot using this ordering and matrix `merge` will not have crossings of the branches.

`labels`: labels for each of the objects being clustered.

`call`: the call which produced the result.

`method`: the cluster method that has been used (e-distance).

`dist.method`: the distance that has been used to create `dst`.

Author(s)

Maria L. Rizzo (`mrizzo @ bgnet.bgsu.edu`) and Gabor J. Szekely (`gabors @ bgnet.bgsu.edu`)

References

Szekely, G. J. and Rizzo, M. L. (2005) Hierarchical Clustering via Joint Between-Within Distances: Extending Ward's Minimum Variance Method, *Journal of Classification* 22(2) 151-183.

<http://dx.doi.org/10.1007/s00357-005-0012-9>

Szekely, G. J. and Rizzo, M. L. (2004) Testing for Equal Distributions in High Dimension, *InterStat*, November (5).

Szekely, G. J. (2000) Technical Report 03-05: \mathcal{E} -statistics: Energy of Statistical Samples, Department of Mathematics and Statistics, Bowling Green State University.

See Also

`edist` `ksample.e` `eqdist.etest` `hclust`

Examples

```
## Not run:

library(cluster)
data(animals)
plot(energy.hclust(dist(animals)))

## End(Not run)

data(USArrests)
```

```

ecl <- energy.hclust(dist(USArrests))
print(ecl)
plot(ecl)
cutree(ecl, k=3)
cutree(ecl, h=150)

## compare performance of e-clustering, Ward's method, group average method
## when sampled populations have equal means: n=200, d=5, two groups
z <- rbind(matrix(rnorm(1000), nrow=200), matrix(rnorm(1000, 0, 5), nrow=200))
g <- c(rep(1, 200), rep(2, 200))
d <- dist(z)
e <- energy.hclust(d)
a <- hclust(d, method="average")
w <- hclust(d^2, method="ward")
list("E" = table(cutree(e, k=2) == g), "Ward" = table(cutree(w, k=2) == g),
     "Avg" = table(cutree(a, k=2) == g))

```

eqdist.etest

Multisample E-statistic (Energy) Test of Equal Distributions

Description

Performs the nonparametric multisample E-statistic (energy) test for equality of multivariate distributions.

Usage

```

eqdist.etest(x, sizes, distance = FALSE, R = 999)
eqdist.e(x, sizes, distance = FALSE)

```

Arguments

x	data matrix of pooled sample
sizes	vector of sample sizes
distance	logical: if TRUE, first argument is a distance matrix
R	number of bootstrap replicates

Details

The k-sample multivariate \mathcal{E} -test of equal distributions is performed. The statistic is computed from the original pooled samples, stacked in matrix x where each row is a multivariate observation, or the corresponding distance matrix. The first $sizes[1]$ rows of x are the first sample, the next $sizes[2]$ rows of x are the second sample, etc.

The test is implemented by nonparametric bootstrap, an approximate permutation test with R replicates. For large samples it is more efficient if x contains the data matrix rather than the distances.

The function `eqdist.e` returns the test statistic only; it simply passes the arguments through to `eqdist.etest` with `R = 0`. For computing the statistic only (no test), `ksample.e` is usually faster.

The definition of the multisample \mathcal{E} -statistic is given in the `ksample.e` documentation.

Value

A list with class `htest` containing

<code>method</code>	description of test
<code>statistic</code>	observed value of the test statistic
<code>p.value</code>	approximate p-value of the test
<code>data.name</code>	description of data

`eqdist.e` returns test statistic only.

Note

The pairwise e-distances between samples can be conveniently computed by the `edist` function, which returns a `dist` object. The function `ksample.e` computes the test statistic without storing the distances.

Author(s)

Maria L. Rizzo (`mrizzo @ bgnet.bgsu.edu`) and Gabor J. Szekely (`gabors @ bgnet.bgsu.edu`)

References

Szekely, G. J. and Rizzo, M. L. (2004) Testing for Equal Distributions in High Dimension, *InterStat*, November (5).

Szekely, G. J. (2000) Technical Report 03-05: \mathcal{E} -statistics: Energy of Statistical Samples, Department of Mathematics and Statistics, Bowling Green State University.

See Also

`ksample.e`, `edist` `energy` `hclust`

Examples

```
data(iris)

## test if the 3 varieties of iris data (d=4) have equal distributions
eqdist.etest(iris[,1:4], c(50,50,50), R = 199)
```

indep.etest

Energy Statistic Test of Independence

Description

Deprecated: use `indep.test` with `method = mvI`. Computes a multivariate nonparametric E-statistic and test of independence.

Usage

```
indep.e(x, y)
indep.etest(x, y, R=199)
```

Arguments

<code>x</code>	matrix: first sample, observations in rows
<code>y</code>	matrix: second sample, observations in rows
<code>R</code>	number of replicates

Details

Computes the coefficient \mathcal{I} and performs a nonparametric \mathcal{E} -test of independence. The test decision is obtained via bootstrap, with `R` replicates. The sample sizes (number of rows) of the two samples must agree, and samples must not contain missing values. The statistic $\mathcal{E} = n\mathcal{I}^2$ is a ratio of V-statistics based on interpoint distances $\|x_i - y_j\|$. See the reference below for details.

Value

The sample coefficient \mathcal{I} is returned by `indep.e`. The function `indep.etest` returns a list with class `htest` containing

<code>method</code>	description of test
<code>statistic</code>	observed value of the coefficient \mathcal{I}
<code>p.value</code>	approximate p-value of the test
<code>data.name</code>	description of data

Author(s)

Maria L. Rizzo <mrizzo @ bgnnet.bgsu.edu> and Gabor J. Szekely <gabors @ bgnnet.bgsu.edu>

References

Bakirov, N.K., Rizzo, M.L., and Szekely, G.J. (2006), A Multivariate Nonparametric Test of Independence, *Journal of Multivariate Analysis* 93/1, 58-80,
<http://dx.doi.org/10.1016/j.jmva.2005.10.005>

Examples

```

## Not run:
## independent univariate data
x <- sin(runif(30, 0, 2*pi) * 2)
y <- sin(runif(30, 0, 2*pi) * 4)
indep.etest(x, y, R = 99)

## dependent univariate data
u <- runif(30, 0, 2*pi)
x <- sin(2 * u)
y <- sin(3 * u)
indep.etest(x, y, R = 99)

u <- runif(50, 0, 2*pi)
x <- sin(2 * u)
y <- sin(4 * u)
indep.etest(x, y, R = 99)

## independent multivariate data
x <- matrix(rnorm(60), nrow=20, ncol=3)
y <- matrix(rnorm(40), nrow=20, ncol=2)
indep.e(x, y)
indep.etest(x, y, R = 99)

## independent bivariate data
x <- matrix(rnorm(50), nrow=25, ncol=2)
y <- matrix(rnorm(50), nrow=25, ncol=2)
indep.e(x, y)
indep.etest(x, y, R = 99)

## dependent bivariate data
library(MASS)
Sigma <- matrix(c(1, .5, .5, 1), 2, 2)
x <- mvrnorm(30, c(0, 0), Sigma)
indep.etest(x[,1], x[,2], R = 99)

## dependent multivariate data
Sigma <- matrix(c(1, .1, 0, 0, 1, 0, 0, .1, 1), 3, 3)
x <- mvrnorm(30, c(0, 0, 0), diag(3))
y <- mvrnorm(30, c(0, 0, 0), Sigma) * x
indep.etest(x, y, R = 99)
## End(Not run)

```

indep.test

Energy Statistic Tests of Independence

Description

Computes a multivariate nonparametric test of independence. The default method implements the distance covariance test `dcov.test`.

Usage

```
indep.test(x, y, method = c("dcov", "mvI"), index = 1, R = 199)
```

Arguments

x	matrix: first sample, observations in rows
y	matrix: second sample, observations in rows
method	a character string giving the name of the test
index	exponent on Euclidean distances
R	number of replicates

Details

`indep.test` with the default `method = "dcov"` computes the distance covariance test of independence. `index` is an exponent on the Euclidean distances. Valid choices for `index` are in $(0,2]$, with default value 1 (Euclidean distance). The arguments are passed to the `dcov.test` function. See the help topic `dcov.test` for the description and documentation and also see the reference (2007) below.

`indep.test` with `method = "mvI"` computes the coefficient \mathcal{I}_n and performs a nonparametric \mathcal{E} -test of independence. The arguments are passed to `mvI.test`. The `index` argument is ignored (`index = 1` is applied). See the help topic `mvI.test` and also see the reference (2006) below for details.

The test decision is obtained via bootstrap, with `R` replicates. The sample sizes (number of rows) of the two samples must agree, and samples must not contain missing values.

These energy tests of independence are based on related theoretical results, but different test statistics. The `dcov` method is faster than `mvI` method by approximately a factor of $O(n)$.

Value

`indep.test` returns a list with class `htest` containing

method	description of test
statistic	observed value of the test statistic $n\mathcal{V}_n^2$ or $n\mathcal{I}_n^2$
estimate	\mathcal{V}_n or \mathcal{I}_n
estimates	a vector [<code>dCov(x,y)</code> , <code>dCor(x,y)</code> , <code>dVar(x)</code> , <code>dVar(y)</code>] (method <code>dcov</code>)
replicates	replicates of the test statistic
p.value	approximate p-value of the test
data.name	description of data

Note

As of `energy-1.1-0`, `indep.etest` is deprecated and replaced by `indep.test`, which has methods for two different energy tests of independence. `indep.test` applies the distance covariance test (see `dcov.test`) by default (`method = "dcov"`). The original `indep.etest` applied the independence coefficient \mathcal{I}_n , which is now obtained by `method = "mvI"`.

Author(s)

Maria L. Rizzo <mrizzo @ bgnet.bgsu.edu> and Gabor J. Szekely <gabors @ bgnet.bgsu.edu>

References

Bakirov, N.K., Rizzo, M.L., and Szekely, G.J. (2006), A Multivariate Nonparametric Test of Independence, *Journal of Multivariate Analysis* 93/1, 58-80,

<http://dx.doi.org/10.1016/j.jmva.2005.10.005>

Szekely, G.J., Rizzo, M.L., and Bakirov, N.K. (2007), Measuring and Testing Dependence by Correlation of Distances, *Annals of Statistics*, Vol. 35 No. 6, pp. 2769-2794.

<http://dx.doi.org/10.1214/0090536070000000505>

See Also

[dcov.test](#) [mvI.test](#) [dcov](#) [mvI](#)

Examples

```
## independent multivariate data
x <- matrix(rnorm(60), nrow=20, ncol=3)
y <- matrix(rnorm(40), nrow=20, ncol=2)
indep.test(x, y, method = "dcov", R = 99)
indep.test(x, y, method = "mvI", R = 99)

## dependent multivariate data
library(MASS)
Sigma <- matrix(c(1, .1, 0, 0, 1, 0, 0, .1, 1), 3, 3)
x <- mvrnorm(30, c(0, 0, 0), diag(3))
y <- mvrnorm(30, c(0, 0, 0), Sigma) * x
indep.test(x, y, R = 99)      #dcov method
indep.test(x, y, method = "mvI", R = 99)

## Not run:
## compare the computing time
x <- mvrnorm(50, c(0, 0, 0), diag(3))
y <- mvrnorm(50, c(0, 0, 0), Sigma) * x
set.seed(123)
system.time(indep.test(x, y, method = "dcov", R = 1000))
set.seed(123)
system.time(indep.test(x, y, method = "mvI", R = 1000))

## End(Not run)

detach("package:MASS")
```

ksample.e	<i>E-statistic (Energy Statistic) for Multivariate k-sample Test of Equal Distributions</i>
-----------	---

Description

Returns the E-statistic (energy statistic) for the multivariate k-sample test of equal distributions.

Usage

```
ksample.e(x, sizes, distance = FALSE, ix = 1:sum(sizes))
```

Arguments

x	data matrix of pooled sample
sizes	vector of sample sizes
distance	logical: if TRUE, x is a distance matrix
ix	a permutation of the row indices of x

Details

The k-sample multivariate \mathcal{E} -statistic for testing equal distributions is returned. The statistic is computed from the original pooled samples, stacked in matrix x where each row is a multivariate observation, or from the distance matrix x of the original data. The first `sizes[1]` rows of x are the first sample, the next `sizes[2]` rows of x are the second sample, etc.

The two-sample \mathcal{E} -statistic proposed by Szekely and Rizzo (2004) is the e-distance $e(S_i, S_j)$, defined for two samples S_i, S_j of size n_i, n_j by

$$e(S_i, S_j) = \frac{n_i n_j}{n_i + n_j} [2M_{ij} - M_{ii} - M_{jj}],$$

where

$$M_{ij} = \frac{1}{n_i n_j} \sum_{p=1}^{n_i} \sum_{q=1}^{n_j} \|X_{ip} - X_{jq}\|,$$

$\|\cdot\|$ denotes Euclidean norm, and X_{ip} denotes the p-th observation in the i-th sample. The k-sample \mathcal{E} -statistic is defined by summing the pairwise e-distances over all $k(k-1)/2$ pairs of samples:

$$\mathcal{E} = \sum_{1 \leq i < j \leq k} e(S_i, S_j).$$

Large values of \mathcal{E} are significant.

Value

The value of the multisample \mathcal{E} -statistic corresponding to the permutation `ix` is returned.

Note

The pairwise e-distances between samples can be conveniently computed by the `edist` function, which returns a `dist` object. The function `ksample.e` computes the \mathcal{E} -statistic only. For the test decision, a nonparametric bootstrap test (approximate permutation test) is provided by the function `eqdist.etest`. With the default arguments, `ksample.e` computes the statistic without storing the distance matrix. For the test statistic only, `ksample.e` is usually faster than calling `eqdist.e`, but for a permutation test the method of calculation in `eqdist.etest` computes the replicates much faster.

Author(s)

Maria L. Rizzo <mrizzo @ bgnet.bgsu.edu> and Gabor J. Szekely <gabors @ bgnet.bgsu.edu>

References

Szekely, G. J. and Rizzo, M. L. (2004) Testing for Equal Distributions in High Dimension, *InterStat*, November (5).

Szekely, G. J. (2000) Technical Report 03-05: \mathcal{E} -statistics: Energy of Statistical Samples, Department of Mathematics and Statistics, Bowling Green State University.

See Also

[eqdist.etest](#) [edist](#) [energy.hclust](#)

Examples

```
## compute 3-sample E-statistic for 4-dimensional iris data
data(iris)
ksample.e(iris[,1:4], c(50,50,50))

## compute a 3-sample univariate E-statistic
ksample.e(rnorm(150), c(25,75,50))
```

mvI

Multivariate Independence Coefficient I

Description

Computes a multivariate coefficient of independence.

Usage

```
mvI(x, y)
```

Arguments

x matrix: first sample, observations in rows
y matrix: second sample, observations in rows

Details

Computes the coefficient \mathcal{I}_n . See `mvI.test` and the reference below for more details.

Value

Returns the scalar statistic, independence coefficient \mathcal{I}_n .

Note

As of energy-1.1-0, `indep.e` is deprecated and replaced by `mvI`.

Author(s)

Maria L. Rizzo (`mrizzo @ bgnet.bgsu.edu`) and Gabor J. Szekely (`gabors @ bgnet.bgsu.edu`)

References

Bakirov, N.K., Rizzo, M.L., and Szekely, G.J. (2006), A Multivariate Nonparametric Test of Independence, *Journal of Multivariate Analysis* 93/1, 58-80,
<http://dx.doi.org/10.1016/j.jmva.2005.10.005>

See Also

`indep.test` `mvI.test` `dcov.test` `dcov`

`mvI.test`

Energy Statistic Test of Independence

Description

Computes the multivariate nonparametric E-statistic and test of independence based on independence coefficient \mathcal{I}_n .

Usage

```
mvI.test(x, y, R=199)
```

Arguments

x	matrix: first sample, observations in rows
y	matrix: second sample, observations in rows
R	number of replicates

Details

Computes the coefficient \mathcal{I} and performs a nonparametric \mathcal{E} -test of independence. The test decision is obtained via bootstrap, with R replicates. The sample sizes (number of rows) of the two samples must agree, and samples must not contain missing values. The statistic $\mathcal{E} = n\mathcal{I}^2$ is a ratio of V-statistics based on interpoint distances $\|x_i - y_j\|$. See the reference below for details.

Value

A list with class `htest` containing

<code>method</code>	description of test
<code>statistic</code>	observed value of the test statistic $n\mathcal{I}_n^2$
<code>estimate</code>	\mathcal{I}_n
<code>replicates</code>	replicates of the test statistic
<code>p.value</code>	approximate p-value of the test
<code>data.name</code>	description of data

Author(s)

Maria L. Rizzo (`mrizzo @ bgnet.bgsu.edu`) and Gabor J. Szekely (`gabors @ bgnet.bgsu.edu`)

References

Bakirov, N.K., Rizzo, M.L., and Szekely, G.J. (2006), A Multivariate Nonparametric Test of Independence, *Journal of Multivariate Analysis* 93/1, 58-80,
<http://dx.doi.org/10.1016/j.jmva.2005.10.005>

See Also

[indep.test](#) [mvI.test](#) [dcov.test](#) [dcov](#)

`mvnorm.etest`

E-statistic (Energy) Test of Multivariate Normality

Description

Performs the E-statistic (energy) test of multivariate or univariate normality.

Usage

```
mvnorm.etest(x, R = 999)
mvnorm.e(x)
normal.e(x)
```

Arguments

x	data matrix of multivariate sample, or univariate data vector
R	number of bootstrap replicates

Details

If x is a matrix, each row is a multivariate observation. The data will be standardized to zero mean and identity covariance matrix using the sample mean vector and sample covariance matrix. If x is a vector, the univariate statistic `normal.e(x)` is returned. If the data contains missing values or the sample covariance matrix is singular, NA is returned.

The \mathcal{E} -test of multivariate normality was proposed and implemented by Szekely and Rizzo (2005). The test statistic for d-variate normality is given by

$$\mathcal{E} = n \left(\frac{2}{n} \sum_{i=1}^n E \|y_i - Z\| - E \|Z - Z'\| - \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \|y_i - y_j\| \right),$$

where y_1, \dots, y_n is the standardized sample, Z, Z' are iid standard d-variate normal, and $\|\cdot\|$ denotes Euclidean norm.

The \mathcal{E} -test of multivariate (univariate) normality is implemented by parametric bootstrap with R replicates.

Value

The value of the \mathcal{E} -statistic for univariate normality is returned by `normal.e`. The value of the \mathcal{E} -statistic for multivariate normality is returned by `mvnorm.e`.

`mvnorm.etest` returns a list with class `htest` containing

method	description of test
statistic	observed value of the test statistic
p.value	approximate p-value of the test
data.name	description of data

Author(s)

Maria L. Rizzo (`mrizzo @ bgnet.bgsu.edu`) and Gabor J. Szekely (`gabors @ bgnet.bgsu.edu`)

References

Szekely, G. J. and Rizzo, M. L. (2005) A New Test for Multivariate Normality, *Journal of Multivariate Analysis*, 93/1, 58-80, <http://dx.doi.org/10.1016/j.jmva.2003.12.002>.

Rizzo, M. L. (2002). A New Rotation Invariant Goodness-of-Fit Test, Ph.D. dissertation, Bowling Green State University.

Szekely, G. J. (1989) Potential and Kinetic Energy in Statistics, Lecture Notes, Budapest Institute of Technology (Technical University).

Examples

```
## compute normality test statistics for iris Setosa data
data(iris)
mvnorm.e(iris[1:50, 1:4])
normal.e(iris[1:50, 1])

## test if the iris Setosa data has multivariate normal distribution
mvnorm.etest(iris[1:50,1:4], R = 199)

## test a univariate sample for normality
x <- runif(50, 0, 10)
mvnorm.etest(x, R = 199)
```

poisson.mtest	<i>Mean Distance Test for Poisson Distribution</i>
---------------	--

Description

Performs the mean distance goodness-of-fit test of Poisson distribution with unknown parameter.

Usage

```
poisson.mtest(x, R = 999)
poisson.m(x)
```

Arguments

x	vector of nonnegative integers, the sample data
R	number of bootstrap replicates

Details

The mean distance test of Poissonity was proposed and implemented by Szekely and Rizzo (2004). The test is based on the result that the sequence of expected values $E|X-j|$, $j=0,1,2,\dots$ characterizes the distribution of the random variable X . As an application of this characterization one can get an estimator $\hat{F}(j)$ of the CDF. The test statistic (see [poisson.m](#)) is a Cramer-von Mises type of distance, with M-estimates replacing the usual EDF estimates of the CDF:

$$M_n = n \sum_{j=0}^{\infty} (\hat{F}(j) - F(j; \hat{\lambda}))^2 f(j; \hat{\lambda}).$$

The test is implemented by parametric bootstrap with R replicates.

Value

The function `poisson.m` returns the test statistic. The function `poisson.mtest` returns a list with class `htest` containing

<code>method</code>	Description of test
<code>statistic</code>	observed value of the test statistic
<code>p.value</code>	approximate p-value of the test
<code>data.name</code>	description of data
<code>estimate</code>	sample mean

Author(s)

Maria L. Rizzo <mrizzo@bgnet.bgsu.edu> and Gabor J. Szekely <gabors@bgnet.bgsu.edu>

References

Szekely, G. J. and Rizzo, M. L. (2004) Mean Distance Test of Poisson Distribution, *Statistics and Probability Letters*, 67/3, 241-247. <http://dx.doi.org/10.1016/j.spl.2004.01.005>.

Examples

```
x <- rpois(20, 1)
poisson.m(x)
poisson.mtest(x, R = 199)
```

Index

*Topic **cluster**

edist, 6
energy.hclust, 7

*Topic **htest**

dcov.test, 1
eqdist.etest, 10
indep.etest, 11
indep.test, 13
mvI.test, 18
mvnorm.etest, 19
poisson.mtest, 20

*Topic **multivariate**

dcov.test, 1
distance correlation, 3
edist, 6
energy.hclust, 7
eqdist.etest, 10
indep.etest, 11
indep.test, 13
ksample.e, 15
mvI, 17
mvI.test, 18
mvnorm.etest, 19

*Topic **nonparametric**

dcov.test, 1
edist, 6
eqdist.etest, 10
indep.test, 13
ksample.e, 15
mvI.test, 18

edist, 6, 9, 11, 17
energy.hclust, 7, 7, 11, 17
eqdist.e(*eqdist.etest*), 10
eqdist.etest, 7, 9, 10, 16, 17

indep.e(*indep.etest*), 11
indep.etest, 11
indep.test, 13, 18, 19

ksample.e, 7, 9–11, 15

mvI, 14, 17
mvI.test, 14, 18, 18, 19
mvnorm.e(*mvnorm.etest*), 19
mvnorm.etest, 19

normal.e(*mvnorm.etest*), 19

poisson.m, 21
poisson.m(*poisson.mtest*), 20
poisson.mtest, 20

DCOR, 3
DCOR(*distance correlation*), 3
dcor, 3
dcor(*distance correlation*), 3
dcov, 2, 3, 14, 18, 19
dcov(*distance correlation*), 3
dcov.test, 1, 4, 5, 13, 14, 18, 19
distance correlation, 3
distance covariance(*dcov.test*), 1