Package ‘energy’

February 23, 2014

Title E-statistics (energy statistics)

Version 1.6.1

Date 2014-02-22

Author Maria L. Rizzo and Gabor J. Szekely

Description E-statistics (energy) tests and statistics for comparing distributions: multivariate normality, multivariate distance components and 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@bgsu.edu>

Imports boot

License GPL (>= 2)

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-02-23 08:11:16

R topics documented:

`energy-package` *E-statistics (energy statistics)*

Description

Description: E-statistics (energy) tests and statistics for comparing distributions: multivariate normality, multivariate distance components and 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.

Author(s)

Maria L. Rizzo and Gabor J. Szekely

References

G. J. Szekely and M. L. Rizzo (2013). Energy statistics: A class of statistics based on distances, *Journal of Statistical Planning and Inference*, http://dx.doi.org/10.1016/j.jspi.2013.03.018

`dcor.ttest` *Distance Correlation t-Test*

Description

Distance correlation t-test of multivariate independence.

Usage

```
dcor.ttest(x, y, distance=FALSE)
dcor.t(x, y, distance=FALSE)
bcdcor(x, y, distance=FALSE)
```

Arguments

`x`	data or distances of first sample
`y`	data or distances of second sample
`distance`	logical: TRUE if x and y are distances

Details

`dcor.ttest` performs a nonparametric t-test of multivariate independence in high dimension (dimension is close to or larger than sample size). The distribution of the test statistic is approximately Student t with $n(n-3)/2-1$ degrees of freedom and for $n \geq 10$ the statistic is approximately distributed as standard normal.

`dcor.t` returns the t statistic and `bcdcor` returns the bias corrected distance correlation statistic.

The sample sizes (number of rows) of the two samples must agree, and samples must not contain missing values. Arguments `x`, `y` can optionally be `dist` objects or distance matrices (in this case set `distance=TRUE`).

The t statistic is a transformation of a bias corrected version of distance correlation (see SR 2013 for details).

Large values (upper tail) of the t statistic are significant.

Value

`dcor.t` returns the t statistic, `bcdcor` returns the bias corrected dcor statistic, and `dcor.ttest` returns a list with class `htest` containing

`method`	description of test
`statistic`	observed value of the test statistic
`parameter`	degrees of freedom
`estimate`	(bias corrected) dCor(x,y)
`p.value`	p-value of the t-test
`data.name`	description of data

Author(s)

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

References

Szekely, G.J. and Rizzo, M.L. (2013). The distance correlation t-test of independence in high dimension. *Journal of Multivariate Analysis*, Volume 117, pp. 193-213.
http://dx.doi.org/10.1016/j.jmva.2013.02.012

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

Szekely, G.J. and Rizzo, M.L. (2009), Brownian Distance Covariance, *Annals of Applied Statistics*, Vol. 3, No. 4, 1236-1265.
http://dx.doi.org/10.1214/09-AOAS312

See Also

`dcov.test` `dcor` `DCOR`

Examples

```
x <- matrix(rnorm(100), 10, 10)
y <- matrix(runif(100), 10, 10)
dx <- dist(x)
dy <- dist(y)
dcor.t(x, y)
bcdcor(dx, dy, distance=TRUE)
dcor.ttest(x, y)
```

---

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	data or distances of first sample
y	data or distances of second sample
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 permutation bootstrap, with `R` replicates.

The sample sizes (number of rows) of the two samples must agree, and samples must not contain missing values. Arguments `x`, `y` can optionally be `dist` objects; otherwise these arguments are treated as data.

The statistic is $n\mathcal{V}_n^2$ where $\mathcal{V}_n(x,y)$ = dcov(x,y), which is based on interpoint Euclidean distances $\|x_i - x_j\|$. The `index` is an optional exponent on Euclidean distance.

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.

Characterization (2) also holds for powers of Euclidean distance $\|x_i - x_j\|^s$, where $0 < s < 2$, but (2) does not hold when $s = 2$.

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), under independence 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 $\mathrm{n\,dCov}^2 = n\mathcal{V}_n^2$ (not dCov).

Author(s)

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

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`

Szekely, G.J. and Rizzo, M.L. (2009), Brownian Distance Covariance, *Annals of Applied Statistics*, Vol. 3, No. 4, 1236-1265.
`http://dx.doi.org/10.1214/09-AOAS312`

Szekely, G.J. and Rizzo, M.L. (2009), Rejoinder: Brownian Distance Covariance, *Annals of Applied Statistics*, Vol. 3, No. 4, 1303-1308.

See Also

`dcov` `dcor` `DCOR` `dcor.ttest`

Examples

```
x <- iris[1:50, 1:4]
y <- iris[51:100, 1:4]
set.seed(1)
dcov.test(x, y)
set.seed(1)
dcov.test(dist(x), dist(y))  #same thing
set.seed(1)
dcov.test(x, y, index=.5)
set.seed(1)
dcov.test(dist(x), dist(y), index=.5)  #same thing

## Example with dvar=0 (so dcov=0 and pval=1)
x <- rep.int(1, 10)
y <- 1:10
dcov.test(x, y, R=199)
```

---

disco *distance components (DISCO)*

Description

E-statistics DIStance COmponents and tests, analogous to variance components

Usage

```
disco(x, factors, distance, index=1.0, R=0, method=c("disco","discoB","discoF"))
disco.between(x, factors, distance, index=1.0, R=0)
```

Arguments

`x`	data matrix or distance matrix
`factors`	matrix of factor labels or integers (not design matrix)
`distance`	logical, TRUE if x is distance matrix
`index`	exponent on Euclidean distance in (0,2]
`R`	number of replicates for a permutation test
`method`	test statistic

Details

`disco` calculates the distance components decomposition of total dispersion and if R > 0 tests for significance using the test statistic disco "F" ratio (default `method="disco"`), or using the between component statistic (`method="discoB"`), each implemented by permutation test.

In the current release `disco` computes the decomposition for one-way models only.

Value

When `method="discoF"`, `disco` returns a class `disco` object, which is a list containing

`call`	call
`method`	method
`statistic`	vector of observed statistics
`p.value`	vector of p-values
`k`	number of factors
`N`	number of observations
`between`	between-sample distance components
`withins`	one-way within-sample distance components
`within`	within-sample distance component
`total`	total dispersion
`Df.trt`	degrees of freedom for treatments
`Df.e`	degrees of freedom for error
`index`	index (exponent on distance)
`factor.names`	factor names
`factor.levels`	factor levels
`sample.sizes`	sample sizes
`stats`	matrix containing decomposition

When `method="discoB"`, `disco` passes the arguments to `disco.between`, which returns a class `htest` object.

`disco.between` returns a class `htest` object, where the test statistic is the between-sample statistic (proportional to the numerator of the F ratio of the `disco` test.

Note

The current version does all calculations via matrix arithmetic and boot function. Support for more general additive models and a formula interface is under development.

`disco` methods have been added to the cluster distance summary function `edist`, and energy tests for equality of distribution (see `eqdist.etest`).

Author(s)

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

References

M. L. Rizzo and G. J. Szekely (2010). DISCO Analysis: A Nonparametric Extension of Analysis of Variance, Annals of Applied Statistics, Vol. 4, No. 2, 1034-1055.
http://dx.doi.org/10.1214/09-AOAS245

See Also

edist eqdist.e eqdist.etest ksample.e

Examples

```
    ## warpbreaks one-way decompositions
    data(warpbreaks)
    attach(warpbreaks)
    disco(breaks, factors=wool, R=99)

    ## When index=2 for univariate data, we get ANOVA decomposition
    disco(breaks, factors=tension, index=2.0, R=99)
    aov(breaks ~ tension)

    ## Multivariate response
    ## Example on producing plastic film from Krzanowski (1998, p. 381)
    tear <- c(6.5, 6.2, 5.8, 6.5, 6.5, 6.9, 7.2, 6.9, 6.1, 6.3,
              6.7, 6.6, 7.2, 7.1, 6.8, 7.1, 7.0, 7.2, 7.5, 7.6)
    gloss <- c(9.5, 9.9, 9.6, 9.6, 9.2, 9.1, 10.0, 9.9, 9.5, 9.4,
               9.1, 9.3, 8.3, 8.4, 8.5, 9.2, 8.8, 9.7, 10.1, 9.2)
    opacity <- c(4.4, 6.4, 3.0, 4.1, 0.8, 5.7, 2.0, 3.9, 1.9, 5.7,
                 2.8, 4.1, 3.8, 1.6, 3.4, 8.4, 5.2, 6.9, 2.7, 1.9)
    Y <- cbind(tear, gloss, opacity)
    rate <- factor(gl(2,10), labels=c("Low", "High"))

## test for equal distributions by rate
    disco(Y, factors=rate, R=99)
disco(Y, factors=rate, R=99, method="discoB")

    ## Just extract the decomposition table
    disco(Y, factors=rate)$stats

## Compare eqdist.e methods for rate
## disco between stat is half of original when sample sizes equal
eqdist.e(Y, sizes=c(10, 10), method="original")
eqdist.e(Y, sizes=c(10, 10), method="discoB")

    ## The between-sample distance component
    disco.between(Y, factors=rate)
```

`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`	data or distances of first sample
`y`	data or distances of second sample
`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. Arguments `x`, `y` can optionally be `dist` objects; otherwise these arguments are treated as data.

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 $0 \le \mathcal{R} \le 1$, 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

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

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

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

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

`dCov`	sample distance covariance
`dCor`	sample distance correlation
`dVarX`	distance variance of x sample
`dVarY`	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 @ bgsu.edu`> and Gabor J. Szekely

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`

Szekely, G.J. and Rizzo, M.L. (2009), Brownian Distance Covariance, *Annals of Applied Statistics*, Vol. 3, No. 4, 1236-1265.
`http://dx.doi.org/10.1214/09-AOAS312`

Szekely, G.J. and Rizzo, M.L. (2009), Rejoinder: Brownian Distance Covariance, *Annals of Applied Statistics*, Vol. 3, No. 4, 1303-1308.

See Also

`dcov.test` `dcor.ttest`

Examples

```
x <- iris[1:50, 1:4]
y <- iris[51:100, 1:4]
dcov(x, y)
dcov(dist(x), dist(y))  #same thing

## C implementation
dcov(x, y, 1.5)
dcor(x, y, 1.5)
.dcov(dist(x), dist(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,
        method = c("cluster","discoB","discoF"))
```

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 in (0,2]
`method`	how to weight the statistics

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 default method `cluster` summarizes the e-distances between clusters in a table. The e-distance between two clusters C_i, C_j of size n_i, n_j proposed by Szekely and Rizzo (2005) 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, $\alpha =$ `alpha`, and X_{ip} denotes the p-th observation in the i-th cluster. The exponent `alpha` should be in the interval (0,2].

The coefficient $\frac{n_i n_j}{n_i + n_j}$ is one-half of the harmonic mean of the sample sizes. The `discoB` and `discoF` methods are related but different ways of summarizing the pairwise differences between samples. The `disco` methods apply the coefficient $\frac{n_i n_j}{2N}$ where N is the total number of observations. This weights each (i,j) statistic by sample size relative to N. See the `disco` topic for more details.

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. The `method` attribute of the distance object is assigned a value of type, index.

Author(s)

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

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`

M. L. Rizzo and G. J. Szekely (2010). DISCO Analysis: A Nonparametric Extension of Analysis of Variance, Annals of Applied Statistics, Vol. 4, No. 2, 1034-1055.
`"http://dx.doi.org/10.1214/09-AOAS245"`

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 disco

Examples

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

## pairwise disco statistics
edist(iris[,1:4], c(50,50,50), method="discoF")  #F ratios



## 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 `dist` object, or a distance matrix produced by `dist`, or lower triangle of distance matrix as vector in column order. If `dst` 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 @ bgsu.edu`> and Gabor J. Szekely

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,
    method=c("original","discoB","discoF"), R = 999)
eqdist.e(x, sizes, distance = FALSE,
    method=c("original","discoB","discoF"))
ksample.e(x, sizes, distance = FALSE,
    method=c("original","discoB","discoF"), ix = 1:sum(sizes))
```

Arguments

`x`	data matrix of pooled sample
`sizes`	vector of sample sizes
`distance`	logical: if TRUE, first argument is a distance matrix
`method`	use original (default) or distance components (discoB, discoF)
`R`	number of bootstrap replicates
`ix`	a permutation of the row indices of x

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.

The function `eqdist.e` returns the test statistic only; it simply passes the arguments through to `eqdist.etest` with `R = 0`.

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 original (default method) k-sample $\mathcal{E}$-statistic is defined by summing the pairwise e-distances over all $k(k-1)/2$ pairs of samples:

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

Large values of $\mathcal{E}$ are significant.

The `discoB` method computes the between-sample disco statistic. For a one-way analysis, it is related to the original statistic as follows. In the above equation, the weights $\frac{n_i n_j}{n_i + n_j}$ are replaced with

$$\frac{n_i + n_j}{2N} \frac{n_i n_j}{n_i + n_j} = \frac{n_i n_j}{2N}$$

where N is the total number of observations: $N = n_1 + ... + n_k$.

The discoF method is based on the disco F ratio, while the discoB method is based on the between sample component.

Also see disco and disco.between functions.

Value

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

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.

Author(s)

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

References

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

M. L. Rizzo and G. J. Szekely (2010). DISCO Analysis: A Nonparametric Extension of Analysis of Variance, Annals of Applied Statistics, Vol. 4, No. 2, 1034-1055.
"http://dx.doi.org/10.1214/09-AOAS245"

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, disco, disco.between, 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)

## example that uses method="disco"
 x <- matrix(rnorm(100), nrow=20)
 y <- matrix(rnorm(100), nrow=20)
 X <- rbind(x, y)
```

```
d <- dist(X)

# should match edist default statistic
set.seed(1234)
eqdist.etest(d, sizes=c(20, 20), distance=TRUE, R = 199)

# comparison with edist
edist(d, sizes=c(20, 10), distance=TRUE)

# for comparison
g <- as.factor(rep(1:2, c(20, 20)))
set.seed(1234)
disco(d, factors=g, distance=TRUE, R=199)

# should match statistic in edist method="discoB", above
set.seed(1234)
disco.between(d, factors=g, distance=TRUE, 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

`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

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

`method`	description of test
`statistic`	observed value of the coefficient $\mathcal{I}$
`p.value`	approximate p-value of the test
`data.name`	description of data

Author(s)

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

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 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 references 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 [dCov(x,y), dCor(x,y), dVar(x), dVar(y)] (method dcov)
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 @ bgsu.edu> and Gabor J. Szekely

References

Szekely, G.J. and Rizzo, M.L. (2009), Brownian Distance Covariance, *Annals of Applied Statistics*, Vol. 3 No. 4, pp. 1236-1265. (Also see discussion and rejoinder.)
http://dx.doi.org/10.1214/09-AOAS312

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

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

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)

## Not run:
## dependent multivariate data
if (require(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)
 }

## End(Not run)

## 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)
```

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)
mvI(x, y)
```

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

`mvI` returns the statistic. `mvI.test` returns a list with class `htest` containing

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

Note

Historically this is the first energy test of independence. The distance covariance test `dcov.test`, distance correlation `dcor`, and related methods are more recent (2007,2009). The distance covariance test is faster and has different properties than `mvI.test`. Both methods are based on a population independence coefficient that characterizes independence and both tests are statistically consistent.

Author(s)

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

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

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

where $y_1, \ldots, y_n$ 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 @ bgsu.edu`> and Gabor J. Szekely

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` *Mean Distance Test for Poisson Distribution*

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,... 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

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

Author(s)

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

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