Package: biotools (via r-universe)

September 4, 2024

Type Package

Title Tools for Biometry and Applied Statistics in Agricultural Science

Version 4.2

Date 2021-08-06

LazyLoad yes

LazyData yes

Author Anderson Rodrigo da Silva [aut, cre] (<https://orcid.org/0000-0003-2518-542X>)

Maintainer Anderson Rodrigo da Silva <anderson.agro@hotmail.com>

Depends MASS

Imports utils, stats, graphics, boot, grDevices, datasets

Suggests soilphysics, tkrplot, tcltk, rpanel, lattice, SpatialEpi, knitr, rmarkdown

Description Tools designed to perform and evaluate cluster analysis (including Tocher's algorithm), discriminant analysis and path analysis (standard and under collinearity), as well as some useful miscellaneous tools for dealing with sample size and optimum plot size calculations. A test for seed sample heterogeneity is now available. Mantel's permutation test can be found in this package. A new approach for calculating its power is implemented. biotools also contains tests for genetic covariance components. Heuristic approaches for performing non-parametric spatial predictions of generic response variables and spatial gene diversity are implemented.

License GPL (>= 2)

VignetteBuilder knitr

URL https://arsilva87.github.io/biotools/

BugReports https://github.com/arsilva87/biotools/issues

Repository https://arsilva87.r-universe.dev

RemoteUrl https://github.com/arsilva87/biotools

RemoteRef HEAD

RemoteSha 85dbdd648b7635070af5f7d5f8eb528f375939a2

Contents

biotools-package . . . 2
aer . . . 4
boxM . . . 5
brazil . . . 6
confusionmatrix . . . 6
cov2pcov . . . 7
creategroups . . . 8
D2.disc . . . 9
D2.dist . . . 10
distClust . . . 12
findSubsample . . . 12
fitplotsize . . . 14
garlicdist . . . 15
gencovtest . . . 15
maize . . . 18
mantelPower . . . 19
mantelTest . . . 20
moco . . . 21
multcor.test . . . 23
optimumplotsize . . . 24
pathanalysis . . . 25
peppercorr . . . 26
raise.matrix . . . 27
samplesize . . . 27
sHe . . . 29
singh . . . 31
tocher . . . 33

Index **35**

`biotools-package` *Tools for Biometry and Applied Statistics in Agricultural Science*

Description

Tools designed to perform and work with cluster analysis (including Tocher's algorithm), discriminant analysis and path analysis (standard and under collinearity), as well as some useful miscellaneous tools for dealing with sample size and optimum plot size calculations. Mantel's permutation test can be found in this package. A new approach for calculating its power is implemented. biotools also contains the new tests for genetic covariance components. An approach for predicting spatial gene diversity is implemented.

Details

Package:	biotools
Type:	Package
Version:	5.0
Date:	2019-01-29
License:	GPL (>= 2)

Note

biotools is an ongoing project. Any and all criticism, comments and suggestions are welcomed.

Author(s)

Anderson Rodrigo da Silva

Maintainer: Anderson Rodrigo da Silva <anderson.agro@hotmail.com>

References

Carvalho, S.P. (1995) *Metodos alternativos de estimacao de coeficientes de trilha e indices de selecao, sob multicolinearidade*. Ph.D. Thesis, Federal University of Vicosa (UFV), Vicosa, MG, Brazil.

Cruz, C.D.; Ferreira, F.M.; Pessoni, L.A. (2011) *Biometria aplicada ao estudo da diversidade genetica*. Visconde do Rio Branco: Suprema.

Lessman, K. J. & Atkins, R. E. (1963). Optimum plot size and relative efficiency of lattice designs for grain sorghum yield tests. *Crop Sci*., 3:477-481.

Mahalanobis, P. C. (1936) On the generalized distance in statistics. *Proceedings of The National Institute of Sciences of India*, 12:49-55.

Manly, B.F.J. (2004) *Multivariate statistical methods*: a primer. CRC Press.

Meier, V. D. & Lessman, K. J. (1971) Estimation of optimum field plot shape and size for testing yield in Crambe abyssinia Hochst. *Crop Sci*., 11:648-650.

Morrison, D.F. (1976) *Multivariate Statistical Methods*.

Rao, R.C. *Advanced statistical methods in biometric research*. New York: John Wiley & Sons, 1952.

Sharma, J.R. (2006) *Statistical and biometrical techniques in plant breeding*. Delhi: New Age International.

da Silva, A.R.; Malafaia, G.; Menezes, I.P.P. (2017) biotools: an R function to predict spatial gene diversity via an individual-based approach. *Genetics and Molecular Research*, 16: gmr16029655.

Silva, A.R. & Dias, C.T.S. (2013) A cophenetic correlation coefficient for Tocher's method. *Pesquisa Agropecuaria Brasileira*, 48:589-596.

Silva et al. (2013) Path analysis in multicollinearity for fruit traits of pepper. *Idesia*, 31:55-60.

Singh, D. (1981) The relative importance of characters affecting genetic divergence. *Indian Journal Genetics & Plant Breeding*, 41:237-245.

aer *Apparent Error Rate*

Description

A function to calculate the apparent error rate of two classification vectors, i.e., the proportion of observed cases incorrectly predicted. It can be useful for evaluating discriminant analysis or other classification systems.

$$aer = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$$

Usage

```
aer(obs, predict)
```

Arguments

obs	a vector containing the observed classes.
predict	a vector with the same length of obs containing the predicted classes.

Value

The apparent error rate, a number between 0 (no agreement) and 1 (thorough agreement).

Author(s)

Anderson Rodrigo da Silva <anderson.agro@hotmail.com>

See Also

confusionmatrix, lda

Examples

```
data(iris)
da <- lda(Species ~ ., data = iris)
pred <- predict(da, dimen = 1)
aer(iris$Species, pred$class)

# End (not run)
```

boxM *Box's M-test*

Description

It performs the Box's M-test for homogeneity of covariance matrices obtained from multivariate normal data according to one classification factor. The test is based on the chi-square approximation.

Usage

```
boxM(data, grouping)
```

Arguments

data	a numeric data.frame or matrix containing n observations of p variables; it is expected that $n > p$.
grouping	a vector of length n containing the class of each observation; it is usualy a factor.

Value

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

statistic	an approximated value of the chi-square distribution.
parameter	the degrees of freedom related of the test statistic in this case that it follows a Chi-square distribution.
p.value	the p-value of the test.
cov	a list containing the within covariance matrix for each level of grouping.
pooled	the pooled covariance matrix.
logDet	a vector containing the natural logarithm of each matrix in cov.
data.name	a character string giving the names of the data.
method	the character string "Box's M-test for Homogeneity of Covariance Matrices".

Author(s)

Anderson Rodrigo da Silva <anderson.agro@hotmail.com>

References

Morrison, D.F. (1976) *Multivariate Statistical Methods*.

Examples

```
data(iris)
boxM(iris[, -5], iris[, 5])

# End (not run)
```

brazil *Brazil Grid*

Description

Lat/Long coordinates within Brazil's limits.

Usage

```
data("brazil")
```

Format

A data frame with 17141 observations on the following 2 variables.

`x` a numeric vector (longitude)

`y` a numeric vector (latitude)

Examples

```
data(brazil)
plot(brazil, cex = 0.1, col = "gray")
```

confusionmatrix *Confusion Matrix*

Description

A function to compute the confusion matrix of two classification vectors. It can be useful for evaluating discriminant analysis or other classification systems.

Usage

```
confusionmatrix(obs, predict)
```

Arguments

`obs`	a vector containing the observed classes.
`predict`	a vector with the same length of `obs` containing the predicted classes.

Value

A square matrix containing the number of objects in each class, observed (rows) and predicted (columns). Diagonal elements refers to agreement of `obs` and `predict`.

Author(s)

Anderson Rodrigo da Silva <anderson.agro@hotmail.com>

See Also

aer, lda

Examples

```
data(iris)
da <- lda(Species ~ ., data = iris)
pred <- predict(da, dimen = 1)
confusionmatrix(iris$Species, pred$class)

# End (not run)
```

---

cov2pcov *Partial Covariance Matrix*

Description

Compute a matrix of partial (co)variances for a group of variables with respect to another.

Take Σ as the covariance matrix of dimension p. Now consider dividing Σ into two groups of variables. The partial covariance matrices are calculate by:

$$\Sigma_{11.2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$$

$$\Sigma_{22.1} = \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}$$

Usage

```
cov2pcov(m, vars1, vars2 = seq(1, ncol(m))[-vars1])
```

Arguments

m	a square numeric matrix.
vars1	a numeric vector indicating the position (rows or columns in m) of the set of variables at which to compute the partial covariance matrix.
vars2	a numeric vector indicating the position (rows or columns in m) of the set of variables at which to adjust the partial covariance matrix.

Value

A square numeric matrix.

Author(s)

Anderson Rodrigo da Silva <anderson.agro at hotmail.com>

See Also

cov

Examples

```
(Cl <- cov(longley))
cov2pcov(Cl, 1:2)

# End (Not run)
```

---

createGroups *Creating Homogeneous Groups*

Description

A function to create homogeneous groups of named objects according to an objective function evaluated at a covariate. It can be useful to design experiments which contain a fixed covariate factor.

Usage

```
createGroups(x, ngroups, sizes, fun = mean, tol = 0.01, maxit = 200)
```

Arguments

x	a numeric vector of a covariate at which to evaluate the objective function.
ngroups	the number of groups to create.
sizes	a numeric vector of length equal to ngroups containing the group sizes.
fun	the objective function, i.e., to create groups with similar fun; default is mean.
tol	the tolerance level to define the groups as homogenenous; see details.
maxit	the maximum number of iterations; default is 200.

Details

createGroups uses a tol value to evaluate the following statistic: $h = \sum_j^{ngroups} abs(t_{j+1} - t_j)/ngroups$, where $t_j = fun(group_j)$. If $h \leq tol$, the groups are considered homogeneous.

Value

A list of

covar	a character indicating the name of the covariate.
func	a character indicating the name of the objective function.
val.func	a numeric vector containing the values evaluated by func on each group.
niter	the number of iteration require to achieve convergence.
labels	a list containing the labels of the objects in each group.
groups	a list of named vectors containing the values for the groups

Author(s)

Anderson Rodrigo da Silva <anderson.agro@hotmail.com>

Examples

```
x <- rnorm(10, 1, 0.5)
names(x) <- letters[1:10]
creategroups(x, ngroups = 2, sizes = c(5, 5))
creategroups(x, ngroups = 3, sizes = c(3, 4, 3), tol = 0.05)

# End (not run)
```

D2.disc *Discriminant Analysis Based on Mahalanobis Distance*

Description

A function to perform discriminant analysis based on the squared generalized Mahalanobis distance (D2) of the observations to the center of the groups.

Usage

```
## Default S3 method:
D2.disc(data, grouping, pooled.cov = NULL)
## S3 method for class 'D2.disc'
print(x, ...)
## S3 method for class 'D2.disc'
predict(object, newdata = NULL, ...)
```

Arguments

data	a numeric data.frame or matrix (*n x p*).
grouping	a vector of length *n* containing the class of each observation (row) in data.
pooled.cov	a grouping-pooled covariance matrix (*p x p*). If NULL (default), D2.disc will automatically compute a pooled covariance matrix.
x, object	an object of class "D2.disc".
newdata	numeric data.frame or matrix of observations to be classified. If NULL (default), the input data used as argument in D2.disc will be used.
...	further arguments.

Value

A list of

call	the call which produced the result.
data	numeric matrix; the input data.
D2	a matrix containing the Mahalanobis distances between each row of data and the center of each class of grouping. In addition, the original and the predicted (lowest distance) class are displayed, as well as a chacater vector indicating where the misclassification has occured.
means	a matrix containing the vector of means of each class in grouping.
pooled	the pooled covariance matrix.
confusion.matrix	an object of class confusionmatrix.

Author(s)

Anderson Rodrigo da Silva <anderson.agro@hotmail.com>

References

Manly, B.F.J. (2004) *Multivariate statistical methods*: a primer. CRC Press. (p. 105-106).

Mahalanobis, P.C. (1936) On the generalized distance in statistics. *Proceedings of The National Institute of Sciences of India*, 12:49-55.

See Also

D2.dist, confusionmatrix, lda

Examples

```
data(iris)
(disc <- D2.disc(iris[, -5], iris[, 5]))
first10 <- iris[1:10, -5]
predict(disc, first10)
predict(disc, iris[, -5])$class

# End (not run)
```

D2.dist *Pairwise Squared Generalized Mahalanobis Distances*

Description

Function to calculate the squared generalized Mahalanobis distance between all pairs of rows in a data frame with respect to a covariance matrix. The element of the *i*-th row and *j*-th column of the distance matrix is defined as

$$D_{ij}^2 = (\boldsymbol{x}_i - \boldsymbol{x}_j)' \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}_i - \boldsymbol{x}_j)$$

Usage

```
D2.dist(data, cov, inverted = FALSE)
```

Arguments

data	a data frame or matrix of data (*n x p*).
cov	a variance-covariance matrix (*p x p*).
inverted	logical. If FALSE (default), cov is supposed to be a variance-covariance matrix.

Value

An object of class "dist".

Author(s)

Anderson Rodrigo da Silva <anderson.agro@hotmail.com>

References

Mahalanobis, P. C. (1936) On the generalized distance in statistics. *Proceedings of The National Institute of Sciences of India*, 12:49-55.

See Also

dist, singh

Examples

```
# Manly (2004, p.65-66)
x1 <- c(131.37, 132.37, 134.47, 135.50, 136.17)
x2 <- c(133.60, 132.70, 133.80, 132.30, 130.33)
x3 <- c(99.17, 99.07, 96.03, 94.53, 93.50)
x4 <- c(50.53, 50.23, 50.57, 51.97, 51.37)
x <- cbind(x1, x2, x3, x4)
Cov <- matrix(c(21.112,0.038,0.078,2.01, 0.038,23.486,5.2,2.844,
 0.078,5.2,24.18,1.134, 2.01,2.844,1.134,10.154), 4, 4)
D2.dist(x, Cov)

# End (not run)
```

distClust *Cluster Distance Matrix*

Description

Function to compute a matrix of average distances within and between clusters.

Usage

```
distClust(d, nobj.cluster, id.cluster)
```

Arguments

d	an object of class "dist" containing the distances between objects.
nobj.cluster	a numeric vector containing the numbers of objects per cluster.
id.cluster	a numeric vector for identification of the objects per cluster.

Value

A squared matrix containing distances within (diagonal) and between (off-diagonal) clusters.

Author(s)

Anderson Rodrigo da Silva <anderson.agro@hotmail.com>

See Also

tocher, dist

findSubsample *Finding an Optimized Subsample*

Description

It allows one to find an optimized (minimized or maximized) numeric subsample according to a statistic of interest. For example, it might be of interest to determine a subsample whose standard deviation is the lowest among all of those obtained from all possible subsamples of the same size.

Usage

```
findSubsample(x, size, fun = sd, minimize = TRUE, niter = 10000)
```

Arguments

x	a numeric vector.
size	an integer; the size of the subsample.
fun	an object of class function; the statistic at which to evaluate the subsample.
minimize	logical; if TRUE (default) findSubsample will find a subsample that minimizes stat.
niter	an integer indicating the number of iterations, i.e., the number of subsamples to be selected (without replacement) from the original sample, x. The larger is this number, the more optimized is the subsample to be found, but this also implies in time-consuming.

Value

A list of

dataname	a character.
niter	the number of iterations.
fun	the objective function.
stat	the achieved statistic for the optimized subsample.
criterion	a character indicating the type of optimization.
subsample	a numeric vector; the optimized subsample.
labels	a string containg the labels of the subsample values.

Author(s)

Anderson Rodrigo da Silva <anderson.agro@hotmail.com>

See Also

sample, creategroups

Examples

```
# Example 1
y <- rnorm(40, 5, 2)
findSubsample(x = y, size = 6)

# Example 2
f <- function(x) diff(range(x)) # max(x) - min(x)
findSubsample(x = y, size = 6, fun = f, minimize = FALSE, niter = 20000)

# End (not run)
```

`fitplotsize` *Parameter Estimation of the Plot Size Model*

Description

Function to estimate the parameters of the nonlinear Lessman & Atkins (1963) model for determining the optimum plot size as a function of the experimental coefficient of variation (CV) or as a function of the residual standard error.

$$CV = a * plotsize^{-b}.$$

It creates initial estimates of the parameters a and b by log-linearization and uses them to provide its least-squares estimates via `nls`.

Usage

```
fitplotsize(plotsize, CV)
```

Arguments

`plotsize`	a numeric vector containing estimates of plot size.
`CV`	a numeric vector of experimental coefficient of variation or residual standard error.

Value

A `nls` output.

Author(s)

Anderson Rodrigo da Silva <anderson.agro@hotmail.com>

References

Lessman, K. J. & Atkins, R. E. (1963) Optimum plot size and relative efficiency of lattice designs for grain sorghum yield tests. *Crop Sci*., 3:477-481.

See Also

`optimumplotsize`

Examples

```
ps <- c(1, 2, 3, 4, 6, 8, 12)
cv <- c(35.6, 29, 27.1, 25.6, 24.4, 23.3, 21.6)
out <- fitplotsize(plotsize = ps, CV = cv)
predict(out) # fitted.values
plot(cv ~ ps)
curve(coef(out)[1] * x^(-coef(out)[2]), add = TRUE)
```

```
# End (not run)
```

`garlicdist` *Distances Between Garlic Cultivars*

Description

The data give the squared generalized Mahalanobis distances between 17 garlic cultivars. The data are taken from the article published by Silva & Dias (2013).

Usage

```
data(garlicdist)
```

Format

An object of class "dist" based on 17 objects.

Source

Silva, A.R. & Dias, C.T.S. (2013) A cophenetic correlation coefficient for Tocher's method. *Pesquisa Agropecuaria Brasileira*, 48:589-596.

Examples

```
data(garlicdist)
tocher(garlicdist)

# End (not run)
```

`gencovtest` *Testing Genetic Covariance*

Description

`gencovtest()` tests genetic covariance components from a MANOVA model. Two different approaches can be used: (I) a test statistic that takes into account the genetic and environmental effects and (II) a test statistic that only considers the genetic information. The first type refers to tests based on the mean cross-products ratio, whose distribution is obtained via Monte Carlo simulation of Wishart matrices. The second way of testing genetic covariance refers to tests based upon an adaptation of Wilks' and Pillai's statistics for evaluating independence of two sets of variables. All these tests are described by Silva (2015).

Usage

```
## S3 method for class 'manova'
gencovtest(obj, geneticFactor, gcov = NULL,
 residualFactor = NULL, adjNrep = 1,
 test = c("MCPR", "Wilks", "Pillai"),
 nsim = 9999,
 alternative = c("two.sided", "less", "greater"))
## S3 method for class 'gencovtest'
print(x, digits = 4, ...)
## S3 method for class 'gencovtest'
plot(x, var1, var2, ...)
```

Arguments

obj	an object of class "manova".
geneticFactor	a character indicating the genetic factor from which to test covariance components. It must be declared as a factor in the manova object.
gcov	optional; a matrix containing estimates of genetic covariances to be tested. If NULL (default), an estimate is obtained via method of moments.
residualFactor	optional; a character indicating a source in the manova model to be used as error term. If NULL (default), the usual term "Residuals" will be used.
adjNrep	a correction index for dealing with unbalanced data. See details.
test	a character indicating the test. It must be on of the following: "MCPR" - the empirical type-I test based on Mean Cross-Products Ratios via Wishart simulation, "Wilks" - a type-II test based on the partial Wilks' Lambda, "Pillai" - a type-II test based on the partial Pillai's statistic.
nsim	the number of Monte Carlo simulations. Used only if test = "MCPR".
alternative	the type of alternative hypothesis. Used only if test = "MCPR". So far, only the option "two.sided" is implemented.
x	an object of class "gencovtest".
digits	the number of digits to be displayed by the print method.
var1	a character of integer indicating one of the two response variable or its position.
var2	a character of integer indicating one of the two response variable or its position.
...	further arguments.

Details

The genetic covariance matrix is currently estimated via method of moments, following the equation:

$$G = (Mg - Me)/(nrep * adjNrep)$$

where Mg and Me are the matrices of mean cross-products associated with the genetic factor and the residuals, respectively; $nrep$ is the number of replications, calculated as the ratio between the total number of observations and the number of levels of the genetic factor; $adjNrep$ is supposed to adjust nrep, specially when estimating G from unbalanced data.

Value

An object of class `gencovtest`, a list of

`gcov`	a p-dimensional square matrix containing estimates of the genetic covariances.
`gcor`	a p-dimensional square matrix containing estimates of the genetic correlations.
`test`	the test (as input).
`statistics`	a p-dimensional square matrix containing the `test` statistics. If `test = "MCPR"` the mean cross-products ratios are computed; if `test = "Wilks"` the Wilks' Lambda is; and `test = "Pillai"` results on Pillai's Tn.
`p.values`	a p-dimensional square matrix containing the associated p-values.
`alternative`	the type of alternative hypothesis (as input).
`X2`	a p-dimensional square matrix containing the Chi-square (D.f. = 1) approximation for Wilks's and Pillai's statistics. Stored only if one of these two tests is chosen.
`simRatio`	an array consisting of `nsim` p-dimensional matrices containing the simulated mean cross-products ratios.
`dfg`	the number of degrees of freedom associated with the genetic factor.
`dfe`	the number of degrees of freedom associated with the residual term.

Warning

When using the MCPR test, be aware that `dfg` should be equal or greater than the number of variables (p). Otherwise the simulation of Wishart matrices may not be done.

A collinearity diagnosis is carried out using the condition number (CN), for the inferences may be affected by the quality of G. Thus, if CN > 100, a warning message is displayed.

Author(s)

Anderson Rodrigo da Silva <anderson.agro@hotmail.com>

References

Silva, A.R. (2015) *On Testing Genetic Covariance*. LAP Lambert Academic Publishing. ISBN 3659716553

See Also

`manova`

Examples

```
# MANOVA
data(maize)
M <- manova(cbind(NKPR, ED, CD, PH) ~ family + env, data = maize)
summary(M)

# Example 1 - MCPR
```

```
t1 <- gencovtest(obj = M, geneticFactor = "family")
print(t1)
plot(t1, "ED", "PH")

# Example 2 - Pillai
t2 <- gencovtest(obj = M, geneticFactor = "family", test = "Pillai")
print(t2)
plot(t2, "ED", "PH")

# End (not run)
```

`maize` *Maize Data*

Description

Data from and experiment with five maize families carried out in randomized block design, with four replications (environments).

Usage

```
data("maize")
```

Format

A data frame with 20 observations on the following 6 variables.

`NKPR` a numeric vector containing values of Number of Kernels Per cob Row.

`ED` a numeric vector containing values of Ear Diameter (in cm).

`CD` a numeric vector containing values of Cob Diameter (in cm).

`PH` a numeric vector containing values of Plant Heigth (in m).

`family` a factor with levels `1` `2` `3` `4` `5`

`env` a factor with levels `1` `2` `3` `4`

Examples

```
data(maize)
str(maize)
summary(maize)
```

`mantelPower` *Power of Mantel's Test*

Description

Power calculation of Mantel's permutation test.

Usage

```
mantelPower(obj, effect.size = seq(0, 1, length.out = 50), alpha = 0.05)
```

Arguments

`obj`	an object of class "mantelTest". See `mantelTest`.
`effect.size`	numeric; the effect size specifying the alternative hypothesis.
`alpha`	numeric; the significance level at which to compute the power level.

Value

A data frame containing the effect size and its respective power level.

Author(s)

Anderson Rodrigo da Silva <anderson.agro@hotmail.com>

References

Silva, A.R.; Dias, C.T.S.; Cecon, P.R.; Rego, E.R. (2015). An alternative procedure for performing a power analysis of Mantel's test. *Journal of Applied Statistics*, doi = 10.1080/02664763.2015.1014894

See Also

`mantelTest`

Examples

```
# Mantel test
data(garlicdist)
garlic <- tocher(garlicdist)
coph <- cophenetic(garlic)
mt1 <- mantelTest(garlicdist, coph, xlim = c(-1, 1))

# Power calculation, H1: rho = 0.3
mantelPower(mt1, effect.size = 0.3)

# Power calculation, multiple H1s and different alphas
p01 <- mantelPower(mt1, alpha = 0.01)
```

```
p05 <- mantelPower(mt1, alpha = 0.05)
p10 <- mantelPower(mt1, alpha = 0.10)
plot(p01, type = "l", col = 4)
lines(p05, lty = 2, col = 4)
lines(p10, lty = 3, col = 4)
legend("bottomright", c("0.10", "0.05", "0.01"),
 title = expression(alpha), col = 4, lty = 3:1, cex = 0.8)

# End (Not run)
```

`mantelTest` *Mantel's Permutation Test*

Description

Mantel's permutation test based on Pearson's correlation coefficient to evaluate the association between two distance square matrices.

Usage

```
mantelTest(m1, m2, nperm = 999, alternative = "greater",
 graph = TRUE, main = "Mantel's test", xlab = "Correlation", ...)
```

Arguments

`m1`	an object of class "matrix" or "dist", containing distances among *n* individuals.
`m2`	an object of class "matrix" or "dist", containing distances among *n* individuals.
`nperm`	the number of matrix permutations.
`alternative`	a character specifying the alternative hypothesis. It must be one of "greater" (default), "two.sided" or "less".
`graph`	logical; if TRUE (default), the empirical distribution is plotted.
`main`	opitional; a character describing the title of the graphic.
`xlab`	opitional; a character describing the *x*-axis label.
`...`	further graphical arguments. See `par`.

Value

A list of

`correlation`	numeric; the observed Pearson's correlation between `m1` and `m2`.
`p.value`	numeric; the empirical p-value of the permutation test.
`alternative`	character; the alternative hypothesis used to compute `p.value`.
`nullcor`	numeric vector containing randomized values of correlation, i.e., under the null hypothesis that the true correlation is equal to zero.

Author(s)

Anderson Rodrigo da Silva <anderson.agro@hotmail.com>

References

Mantel, N. (1967). The detection of disease clustering and a generalized regression approach. *Cancer Research*, 27:209–220.

See Also

mantelPower

Examples

```
# Distances between garlic cultivars
data(garlicdist)
garlicdist

# Tocher's clustering
garlic <- tocher(garlicdist)
garlic

# Cophenetic distances
coph <- cophenetic(garlic)
coph

# Mantel's test
mantelTest(garlicdist, coph,
 xlim = c(-1, 1))

# End (Not run)
```

---

moco *Moco Cotton Data*

Description

Data set of...

Usage

```
data("moco")
```

Format

A data frame with 206 observations (sampling points) on the following 20 variables (coordinates and markers).

`Lon` a numeric vector containing values of longitude

`Lat` a numeric vector containing values of latitude

`BNL1434.1` a numeric vector (marker)

`BNL1434.2` a numeric vector

`BNL840.1` a numeric vector

`BNL840.2` a numeric vector

`BNL2496.1` a numeric vector

`BNL2496.2` a numeric vector

`BNL1421.1` a numeric vector

`BNL1421.2` a numeric vector

`BNL1551.1` a numeric vector

`BNL1551.2` a numeric vector

`CIR249.1` a numeric vector

`CIR249.2` a numeric vector

`BNL3103.1` a numeric vector

`BNL3103.2` a numeric vector

`CIR311.1` a numeric vector

`CIR311.2` a numeric vector

`CIR246.1` a numeric vector

`CIR246.2` a numeric vector

Source

...

References

...

Examples

```
data(moco)
str(moco)
```

multcor.test *Pairwise Correlation t-Test*

Description

It performs multiple correlation t-tests from a correlation matrix based on the statistic:

$$t = r * \sqrt{(df/(1 - r^2))}$$

where, in general, $df = n - 2$.

Usage

```
multcor.test(x, n = NULL, Df = NULL,
 alternative = c("two.sided", "less", "greater"), adjust = "none")
```

Arguments

x	a correlation matrix.
n	the number of observations; if NULL (default), the argument Df must be passed.
Df	the number of degrees of freedom of the t statistic; if NULL (default), the argument n must be passed and, in this case, multcor.test considers $Df = n - 2$.
alternative	the alternative hypothesis. It must be one of "two.sided", "greater" or "less". You can specify just the initial letter. "greater" corresponds to positive association,"less" to negative association. The default is "two.sided".
adjust	The adjustment method for multiple tests. It must be one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none" (default). For more information, see p.adjust.

Value

A list with class "multcor.test" containing the following components:

t.values	the t-value calculated for each correlation.
p.values	the p.value for each t-test, adjusted for multiple tests.
p.check	a matrix containing the p.values for each t-test (lower triangular) and a symbol indicating the significance level at which one can to reject the null hypothesis (upper triangular).
adjustemnt	a character indicating the p-value adjustment method.
df	the degrees of freedom of the tests.
alternative	a character indicating the type of alternative hypothesis.
data.name	a character string giving the name of the data.

Author(s)

Anderson Rodrigo da Silva <anderson.agro@hotmail.com>

See Also

cor, cor.test, p.adjust

Examples

```
data(peppercorr)
multcor.test(peppercorr, n = 20)

# End (not run)
```

---

optimumplotsize *Maximum Curvature Point for Optimum Plot Size*

Description

The Meier & Lessman (1971) method to determine the maximum curvature point for optimum plot size as a function of the experimental coefficient of variation.

Usage

```
optimumplotsize(a, b)
```

Arguments

a	a parameter estimate of the plot size model; see fitplotsize.
b	a parameter estimate of the plot size model; see fitplotsize.

Value

The (approximated) optimum plot size value.

Author(s)

Anderson Rodrigo da Silva <anderson.agro@hotmail.com>

References

Meier, V. D. & Lessman, K. J. (1971) Estimation of optimum field plot shape and size for testing yield in Crambe abyssinia Hochst. *Crop Sci*., 11:648-650.

See Also

fitplotsize

Examples

```
ps <- c(1, 2, 3, 4, 6, 8, 12)
cv <- c(35.6, 29, 27.1, 25.6, 24.4, 23.3, 21.6)
out <- fitplotsize(plotsize = ps, CV = cv)
plot(cv ~ ps)
curve(coef(out)[1] * x^(-coef(out)[2]), add = TRUE)
optimumplotsize(a = coef(out)[1], b = coef(out)[2])

# End (not run)
```

`pathanalysis` *Path Analysis, Simple and Under Collinearity*

Description

Function to perform the simple path analysis and the path analysis under collinearity (sometimes called *ridge path analysis*). It computes the direct (diagonal) and indirect (off-diagonal) effects of each explanatory variable over a response one.

Usage

```
pathanalysis(corMatrix, resp.col, collinearity = FALSE)
```

Arguments

`corMatrix`	a correlation matrix.
`resp.col`	an integer value indicating the column in `corMatrix` that corresponds to the response variable.
`collinearity`	logical; if `TRUE`, an external interactive display is used to pass a value, say k, at which to evaluate the system: $(\mathbf{X'X} + \mathbf{I}k)\mathbf{B} = \mathbf{X'Y}$, being $\mathbf{X'X}$ the correlation matrix between explanatory variables, $\mathbf{X'Y}$ the correlation vector between all explanatory variables and the response variable, $\mathbf{B}$ is the vector of path coefficients and k is a value between 0 and 1; default is `FALSE`, i.e., k = 0.

Value

A list of

`coef`	a matrix containing the direct (diagonal) and indirect (off-diagonal) effects of each variable.
`Rsq`	the coefficient of determination.
`ResidualEffect`	the residual effect.
`VIF`	a vector containing the variance inflation factors.
`CN`	the condition number.

Side Effects

If `collinearity = TRUE`, an interactive graphic is displayed for dealing with collinearity.

Author(s)

Anderson Rodrigo da Silva <anderson.agro@hotmail.com>

References

Carvalho, S.P. (1995) *Metodos alternativos de estimacao de coeficientes de trilha e indices de selecao, sob multicolinearidade*. Ph.D. Thesis, Federal University of Vicosa (UFV), Vicosa, MG, Brazil.

Examples

```
data(peppercorr)
pathanalysis(peppercorr, 6, collinearity = FALSE)

# End (not run)
```

`peppercorr` *Correlations Between Pepper Variables*

Description

The data give the correlations between 6 pepper variables. The data are taken from the article published by Silva et al. (2013).

Usage

```
data(peppercorr)
```

Format

An object of class "matrix".

Source

Silva et al. (2013) Path analysis in multicollinearity for fruit traits of pepper. *Idesia*, 31:55-60.

Examples

```
data(peppercorr)
print(peppercorr)

# End (not run)
```

raise.matrix *Raising a Square Matrix to a Power*

Description

`raise.matrix` raises a square matrix to a power by using spectral decomposition.

Usage

```
raise.matrix(x, power = 1)
```

Arguments

`x`	a square matrix.
`power`	numeric; default is 1.

Value

An object of class "matrix".

Author(s)

Anderson Rodrigo da Silva <anderson.agro@hotmail.com>

See Also

`eigen`, `svd`

Examples

```
m <- matrix(c(1, -2, -2, 4), 2, 2)
raise.matrix(m)
raise.matrix(m, 2)

# End (not run)
```

samplesize *Minimum Sample Size*

Description

Function to determine the minimum sample size for calculating a statistic based on its the confidence interval.

Usage

```
samplesize(x, fun, sizes = NULL, lcl = NULL, ucl = NULL,
 nboot = 200, conf.level = 0.95, nrep = 500, graph = TRUE, ...)
```

Arguments

x	a numeric vector.
fun	an objective function at which to evaluate the sample size; see details.
sizes	a numeric vector containing sample sizes; if NULL (default), samplesize creates a vector ranging from 2 to *n*-1.
lcl	the lower confidence limit for the statistic defined in fun; if NULL (default), samplesize estimates lcl based on bootstrap percentile interval.
ucl	the upper confidence limit for the statistic defined in fun; if NULL (default), samplesize estimates ucl based on bootstrap percentile interval.
nboot	the number of bootstrap samples; it is used only if lcl or ucl is NULL.
conf.level	the confidence level for calculating the lcl and ucl; it is used only if lcl or ucl is NULL.
nrep	the resampling (with replacement) number for each sample size in sizes; default is 500.
graph	logical; default is TRUE.
...	further graphical arguments.

Details

If ucl or lcl is NULL, fun must be defined as in boot, i.e., the first argument passed will always be the original data and the second will be a vector of indices, frequencies or weights which define the bootstrap sample. By now, samplesize considers the second argument only as index.

Value

A list of

CI	a vector containing the lower and the upper confidence limit for the statistic evaluated.
pointsOut	a data frame containing the sample sizes (in sizes), the number of points outside the CI (n.out) and the proportion of this number (prop).

Side Effects

If graph = TRUE, a graphic with the dispersion of the estimates for each sample size, as well as the graphic containing the number of points outside the confidence interval for the reference sample.

Author(s)

Anderson Rodrigo da Silva <anderson.agro@hotmail.com>

Examples

```
cv <- function(x, i) sd(x[i]) / mean(x[i]) # coefficient of variation
x = rnorm(20, 15, 2)
cv(x)
samplesize(x, cv)

par(mfrow = c(1, 3), cex = 0.7, las = 1)
samplesize(x, cv, lcl = 0.05, ucl = 0.20)
abline(h = 0.05 * 500, col = "blue") # sample sizes with 5% (or less) out CI

# End (not run)
```

---

sHe *Spatial Analysis of Gene Diversity*

Description

Estimate spatial gene diversity (expected heterozygozity - *He*) through the individual-centred approach by Manel et al. (2007). `sHe()` calculates the unbiased estimate of *He* based on the information of allele frequency obtained from codominant or dominant markers in individuals within a circular moving windows of known radius over the sampling area.

Usage

```
sHe(x, coord.cols = 1:2, marker.cols = 3:4,
 marker.type = c("codominant", "dominant"),
 grid = NULL, latlong2km = TRUE, radius, nmin = NULL)
```

Arguments

`x`	a data frame or numeric matrix containing columns with coordinates of individuals and marker genotyping
`coord.cols`	a vector of integer giving the columns of coordinates in `x`
`marker.cols`	a vector of integer giving the columns of markers in `x`
`marker.type`	a character; the type of molecular marker
`grid`	optional; a two-column matrix containing coordinates over which to predict *He*
`latlong2km`	logical; should coordinates be converted from lat/long format into kilometer-grid based?
`radius`	the radius of the moving window. It must be in the same format as sampling coordinates
`nmin`	optional; a numeric value indicating the minimum number of individuals used to calculate *He*. If is the number of individuals in a certain location is less then `nmin`, `sHe` will consider *He* as zero.

Details

The unbiased estimate of expected heterogygozity (Nei, 1978) is given by:

$$He = (1 - \sum_{i=1}^{n} p_i^2) \frac{2n}{2n-1}$$

where p_i is the frequency of the i-th allele per locus considering the n individuals in a certain location.

Value

A list of

`diversity`	a data frame with the following columns: *coord.x* - the x-axis coordinates of the predicion grid, *coord.y* - the y-axis coordinates of the predicion grid, *n* - the number of individuals in a certain points in the grid, *MaxDist* - the maximum observed distance among these individuals, *uHe* - the unbiased estimate of gene diversity (as expressed above), and *SE* - the standard error of *uHe*.
`mHe`	a matrix containing the estimates of *He* for every marker, on each point of the `grid`.
`locations`	a numeric matrix containing the sampling coordinates, as provides as input.

Warning

Depending on the dimension of `x` and/or `grid`, `sHe()` can be time demanding.

Author(s)

Anderson Rodrigo da Silva <anderson.agro@hotmail.com>

Ivandilson Pessoa Pinto de Menezes <ivan.menezes@ifgoiano.edu.br>

References

da Silva, A.R.; Malafaia, G.; Menezes, I.P.P. (2017) biotools: an R function to predict spatial gene diversity via an individual-based approach. *Genetics and Molecular Research*, **16**: gmr16029655.

Manel, S., Berthoud, F., Bellemain, E., Gaudeul, M., Luikart, G., Swenson, J.E., Waits, L.P., Taberlet, P.; Intrabiodiv Consortium. (2007) A new individual-based spatial approach for identifying genetic discontinuities in natural populations. *Molecular Ecology*, **16**:2031-2043.

Nei, M. (1978) Estimation of average heterozygozity and genetic distance from a small number of individuals. *Genetics*, **89**: 583-590.

See Also

`levelplot`

Examples

```
data(moco)
data(brazil)

# check points
plot(brazil, cex = 0.1, col = "gray")
points(Lat ~ Lon, data = moco, col = "blue", pch = 20)

# using a retangular grid (not passed as input!)
# ex <- sHe(x = moco, coord.cols = 1:2,
# marker.cols = 3:20, marker.type = "codominant",
# grid = NULL, radius = 150)
#ex
# plot(ex, xlab = "Lon", ylab = "Lat")

# A FANCIER PLOT...
# using Brazil's coordinates as prediction grid
# ex2 <- sHe(x = moco, coord.cols = 1:2,
# marker.cols = 3:20, marker.type = "codominant",
# grid = brazil, radius = 150)
# ex2
#
# library(maps)
# borders <- data.frame(x = map("world", "brazil")$x,
# y = map("world", "brazil")$y)
#
# library(latticeExtra)
# plot(ex2, xlab = "Lon", ylab = "Lat",
# xlim = c(-75, -30), ylim = c(-35, 10), aspect = "iso") +
#   latticeExtra::as.layer(xyplot(y ~ x, data = borders, type = "l")) +
#   latticeExtra::as.layer(xyplot(Lat ~ Lon, data = moco))

# End (not run)
```

---

singh *Importance of Variables According to the Singh (1981) Criterion*

Description

A function to calculate the Singh (1981) criterion for importance of variables based on the squared generalized Mahalanobis distance.

$$S_{.j} = \sum_{i=1}^{n-1} \sum_{i'>i}^{n} (x_{ij} - x_{i'j}) * (\boldsymbol{x}_i - \boldsymbol{x}_{i'})' * \boldsymbol{\Sigma}_j^{-1}$$

Usage

```
## Default S3 method:
singh(data, cov, inverted = FALSE)
```

```
## S3 method for class 'singh'
plot(x, ...)
```

Arguments

data	a data frame or matrix of data (*n x p*).
cov	a variance-covariance matrix (*p x p*).
inverted	logical. If FALSE (default), cov is supposed to be a variance-covariance matrix.
x	an object of class "singh".
...	further graphical arguments.

Value

singh returns a matrix containing the Singh statistic, the importance proportion and the cummulative proprtion of each variable (column) in data.

Author(s)

Anderson Rodrigo da Silva <anderson.agro@hotmail.com>

References

Singh, D. (1981) The relative importance of characters affecting genetic divergence. *Indian Journal Genetics & Plant Breeding*, 41:237-245.

See Also

D2.dist

Examples

```
# Manly (2004, p.65-66)
x1 <- c(131.37, 132.37, 134.47, 135.50, 136.17)
x2 <- c(133.60, 132.70, 133.80, 132.30, 130.33)
x3 <- c(99.17, 99.07, 96.03, 94.53, 93.50)
x4 <- c(50.53, 50.23, 50.57, 51.97, 51.37)
x <- cbind(x1, x2, x3, x4)
Cov <- matrix(c(21.112,0.038,0.078,2.01, 0.038,23.486,5.2,2.844,
 0.078,5.2,24.18,1.134, 2.01,2.844,1.134,10.154), 4, 4)
(s <- singh(x, Cov))
plot(s)

# End (not run)
```

tocher *Tocher's Clustering*

Description

tocher performs the Tocher (Rao, 1952) optimization clustering from a distance matrix. The cophenetic distance matrix for a Tocher's clustering can also be computed using the methodology proposed by Silva \& Dias (2013).

Usage

```
## S3 method for class 'dist'
tocher(d, algorithm = c("original", "sequential"))
## S3 method for class 'tocher'
print(x, ...)
## S3 method for class 'tocher'
cophenetic(x)
```

Arguments

d	an object of class "dist".
algorithm	a character indicating the algorithm to be used for clustering objects. It must be one of the two: "original" (default) or "sequential". The latter is the method proposed by Vasconcelos et al. (2007), and sometimes called "modified" Tocher.
x	an object of class "tocher".
...	optional further arguments from print.

Value

An object of class tocher. A list of

call	the call which produced the result.
algorithm	character; the algorithm that has been used as input.
clusters	a list of length *k* (the number of clusters), containing the labels of the objects in d for each cluster.
class	a numeric vector indicating the class (the cluster) of each object in d.
criterion	a numeric vector containing the clustering criteria - the greatest amongst the smallest distances involving each object in d. If algorithm = "original", this vector contains an unique value, i.e., the same criterion is used for every clustering step.
distClust	a matrix of distances within (diagonal) and between (off-diagonal) clusters.
d	the input object.

Warning

Clustering a large number of objects (say 300 or more) can be time demanding.

Author(s)

Anderson Rodrigo da Silva <anderson.agro@hotmail.com>

References

Cruz, C.D.; Ferreira, F.M.; Pessoni, L.A. (2011) *Biometria aplicada ao estudo da diversidade genetica*. Visconde do Rio Branco: Suprema.

Rao, R.C. (1952) *Advanced statistical methods in biometric research*. New York: John Wiley & Sons.

Sharma, J.R. (2006) *Statistical and biometrical techniques in plant breeding*. Delhi: New Age International.

Silva, A.R. & Dias, C.T.S. (2013) A cophenetic correlation coefficient for Tocher's method. *Pesquisa Agropecuaria Brasileira*, 48:589-596.

Vasconcelos, E.S.; Cruz, C.D.; Bhering, L.L.; Resende Junior, M.F.R. (2007) Alternative methodology for the cluster analysis. *Pesquisa Agropecuaria Brasileira*, 42:1421-1428.

See Also

dist, D2.dist, cophenetic, distClust, hclust

Examples

```
# example 1
data(garlicdist)
(garlic <- tocher(garlicdist))
garlic$distClust  # cluster distances

# example 2
data(USArrests)
(usa <- tocher(dist(USArrests)))
usa$distClust

# cophenetic correlation
cophUS <- cophenetic(usa)
cor(cophUS, dist(USArrests))

# using the sequential algorithm
(usa2 <- tocher(dist(USArrests), algorithm = "sequential"))
usa2$criterion

# example 3
data(eurodist)
(euro <- tocher(eurodist))
euro$distClust

# End (not run)
```

Index

∗ **datasets**
  brazil, 6
  garlicdist, 15
  maize, 18
  moco, 21
  peppercorr, 26
∗ **package**
  biotools-package, 2

aer, 4, *7*

biotools *(*biotools-package*)*, 2
biotools-package, 2
boot, *28*
boxM, 5
brazil, 6

confusionmatrix, *4*, 6, *10*
coph.tocher *(*tocher*)*, 33
cophenetic, *34*
cophenetic.tocher *(*tocher*)*, 33
cor, *24*
cor.test, *24*
cov, *8*
cov2pcov, 7
creategroups, 8, *13*

D2.disc, 9
D2.dist, *10*, 10, *32*, *34*
dist, *11, 12*, *34*
distClust, 12, *34*

eigen, *27*

findSubsample, 12
fitplotsize, 14, *24*

garlicdist, 15
gencovtest, 15

hclust, *34*

lda, *4*, *7*, *10*
levelplot, *30*

maize, 18
manova, *17*
mantelPower, 19, *21*
mantelTest, *19*, 20
moco, 21
multcor.test, 23

nls, *14*

optimumplotsize, *14*, 24

p.adjust, *23, 24*
par, *20*
pathanalysis, 25
peppercorr, 26
plot.gencovtest *(*gencovtest*)*, 15
plot.singh *(*singh*)*, 31
predict.D2.disc *(*D2.disc*)*, 9
print.D2.disc *(*D2.disc*)*, 9
print.gencovtest *(*gencovtest*)*, 15
print.tocher *(*tocher*)*, 33

raise.matrix, 27

sample, *13*
samplesize, 27
sHe, 29
singh, *11*, 31
svd, *27*

tocher, *12*, 33