Package ‘ProcMod’

May 12, 2021

Type  Package
Title  Informative Procrustean Matrix Correlation
Version  1.0.8
Author  Eric Coissac, Christelle Gonindard-Melodelima
Maintainer  Eric Coissac <eric.coissac@metabarcoding.org>
License  CeCILL-2
Encoding  UTF-8
LazyData  true
RoxygenNote  7.1.1
Depends  R (>= 3.1.0)
Imports  MASS, permute, Matrix, stats, foreach, Rdpack
Suggests  knitr, rmarkdown, roxygen2, vegan, testthat, ade4, doParallel
RdMacros  Rdpack
Collate  'internals.R' 'procmod_frame.R' 'multivariate.R' 'procmod.R'
          'covls.R' 'corls_test.R' 'procuste.R' 'simulate.R'
NeedsCompilation  no
Repository  CRAN
Date/Publication  2021-05-12 06:52:11 UTC

R topics documented:

  .getPermuteMatrix ................................................. 2
  .procmod_coerce_value ........................................... 3
  .rep_matrix ...................................................... 3
  .Trace ............................................................ 4
  as.data.frame.dist ............................................... 5
  as.procmod_frame ................................................ 6
  bicenter .......................................................... 7
.getPermuteMatrix

Generate permutation matrix according to a schema.

Description

The permutation schema is defined using the ‘how’ function. The implementation of this function is inspired from the VEGAN package and reproduced here to avoid an extra dependency on an hidden vegan function.

Usage

.getPermuteMatrix(permuations, n, strata = NULL)

Arguments

permutations  a list of control values for the permutations as returned by the function how, or the number of permutations required.
n  numeric; the number of observations in the sample set. May also be any object that nobs knows about; see nobs methods.
strata  A factor, or an object that can be coerced to a factor via as.factor, specifying the strata for permutation.

Note

Internal function do not use.
.procmod_coerce_value  

"Internal function coercing the data to a matrix."

Description

Transform the x value into a numeric matrix of the correct size or into a dist object.

Usage

.procmod_coerce_value(x, nrows = 0, contrasts = NULL)

Arguments

x  The data to coerce
nrows  an integer value specifying the number of row of the returned matrix
contrasts  see the contrasts_arg argument of the procmod_frame constructor.

Value

a new numeric matrix with correct size.

Note

Internal function do not use.

Author(s)

Eric Coissac <eric.coissac@metabarcoding.org>
Christelle Gonindard-Melodelima <christelle.gonindard@metabarcoding.org>

.rep_matrix  

"Internal function repeating a matrix."

Description

repeats several times the rows of a matrix to create a new matrix with more rows. The final row count must be a multiple of the initial row count.

Usage

.rep_matrix(x, nrow)

Arguments

x  The matrix to replicate
nrow  an integer value specifying the number of row of the returned matrix
.Trace

Value

a new matrix with the same number of columns but with 'nrow' rows.

Note

Internal function do not use.

Author(s)

Eric Coissac <eric.coissac@metabarcoding.org>
Christelle Gonindard-Melodelima <christelle.gonindard@metabarcoding.org>

Description

The trace of a square matrix is defined as the sum of its diagonal elements.

Usage

.Trace(X)

Arguments

X a square matrix

Value

the trace of X

Note

Internal function do not use.

Author(s)

Eric Coissac
Christelle Gonindard-Melodelima

Examples

m <- matrix(1:16, nrow = 4)
ProcMod::.Trace(m)
as.data.frame.dist

Converts a dist object to a data.frame object.

Description

The created data.frame has a attribute is.dist set to the logical value TRUE.

Usage

```r
## S3 method for class 'dist'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

- **x**: the dist object to be converted
- **row.names**: NULL or a character vector giving the row names for the data frame. Missing values are not allowed.
- **optional**: logical. If TRUE, setting row names and converting column names (to syntactic names: see make.names) is optional. Note that all of R's base package as.data.frame() methods use optional only for column names treatment, basically with the meaning of data.frame(*, check.names = !optional). See also the make.names argument of the matrix method.
- **...**: additional arguments to be passed to or from methods.

Author(s)

Eric Coissac
Christelle Gonindard-Melodelima

Examples

```r
data(bacteria)
bacteria_rel_freq <- sweep(bacteria, 
    1, rowSums(bacteria), 
    
"/")
bacteria_hellinger <- sqrt(bacteria_rel_freq)
bacteria_dist <- dist(bacteria_hellinger)
bdf <- as.data.frame(bacteria_dist)
```
as_procmod_frame

Coerce to a ProcMod Frame.

Description
Conversion methods are proposed for list, matrix and array.

Usage
as_procmod_frame(data, ...)

## S3 method for class 'list'
as_procmod_frame(data, ...)

## S3 method for class 'procmod_frame'
as_procmod_frame(data, ...)

## S3 method for class 'array'
as_procmod_frame(data, ...)

## S3 method for class 'matrix'
as_procmod_frame(data, ...)

Arguments
data a R object to coerce.
... supplementary parameters used in some implementation of that method

Value
a procmod_frame object

Author(s)
Eric Coissac
Christelle Gonindard-Melodelima

Examples
# Builds a list containing two random matrices
m1 <- simulate_matrix(10,20)
m2 <- simulate_matrix(10,30)
l <- list(m1 = m1, m2 = m2)

# Converts the list to a procmod_frame
pmf1 <- as_procmod_frame(l)

# Builds a procmod_frame from a matrix
bicenter

Double centering of a matrix.

Description

colSums and rowSums of the returned matrix are all equal to zero.

Usage

bicenter(m)

Arguments

m a numeric matrix

Details

Inspired from the algorithm described in stackoverflow https://stackoverflow.com/questions/43639063/double-centering-in-r

Value

a numeric matrix centred by rows and columns

Author(s)

Eric Coissac
Christelle Gonindard-Melodelima

Examples

data(bacteria)
bact_bc <- bicenter(bacteria)
sum(rowSums(bact_bc))
sum(colSums(bact_bc))
**corls_test**  
*Monte-Carlo Test on the sum of the singular values of a procustean rotation.*

**Description**

performs a Monte-Carlo Test on the sum of the singular values of a procustean rotation (see `procuste.rtest`).

**Usage**

```r
corls_test(
  ..., 
  permutations = permute::how(nperm = 999), 
  p_adjust_method = "holm"
)
```

**Arguments**

- `...`: the set of matrices or a `procmod_frame` object.
- `permutations`: a list of control values for the permutations as returned by the function `how`, or the number of permutations required.
- `p_adjust_method`: the multiple test correction method used to adjust p values. `p_adjust_method` belongs one of the following values: "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none". The default is set to "holm".

**Author(s)**

Eric Coissac  
Christelle Gonindard-Melodelima

**References**


**See Also**

`p.adjust`

**Examples**

```r
A <- simulate_matrix(10,3)  
B <- simulate_matrix(10,5)  
C <- simulate_correlation(B,10,r2=0.6)

# Computes the correlation matrix  
data <- procmod_frame(A = A, B = B, C = C)
```
**dim.procmod_frame**

`corls_test(data, permutations = 100)`

---

**dim.procmod_frame**  
*Dimensions of a ProcMod Frame.*

**Description**

Dimension 1 is the number of rows (individus) shared by the aggregated matrices. Dimension 2 is the number of aggregated matrices.

**Usage**

```r
## S3 method for class 'procmod_frame'
dim(x)
```

**Arguments**

- `x`  
  a `procmod_frame` object

**Author(s)**

Eric Coissac

Christelle Gonindard-Melodelima

**Examples**

```r
# Builds a procmod_frame with two random matrices
m1 <- simulate_matrix(10,20)
m2 <- simulate_matrix(10,30)
pmf <- procmod_frame(m1 = m1, m2 = m2)
dim(pmf)
```

---

**eukaryotes**  
*DNA metabarcoding Australia South-North Gradient*

**Description**

This data set of five `data.frame` is a simplified version of a full data set describing biodiversity changes along a South-North gradient on the Australian East Coast, from Sidney to North Cap using a DNA metabarcoding approach. The gradient is constituted of 21 locations.
Usage

data(eukaryotes)
data(bacteria)
data(climat)
data(soil)
data(geography)

Format

five data.frame of 21 rows
An object of class data.frame with 21 rows and 2150 columns.
An object of class data.frame with 21 rows and 6 columns.
An object of class data.frame with 21 rows and 12 columns.
An object of class data.frame with 21 rows and 2 columns.

Details

**bacteria** is a 21 x 2150 data.frame describing bacterial community at each one of the 21 locations. Each number is the relative frequency of a molecular operational taxonomy unit (MOTU) at a site after data cleaning and averaging of 135 pontual measures.

**bacteria** is a 21 x 1393 data.frame describing eukariote community at each one of the 21 locations. Each number is the relative frequency of a molecular operational taxonomy unit (MOTU) at a site after data cleaning and averaging of 135 pontual measures.

**climat** is a 21 x 6 data.frame describing climatic conditions at each site using worldclim descriptors (https://www.worldclim.org).

  * Aspect
  * TempSeasonality
  * MaxMonTemp  Max Temperature of Warmest Month
  * MeanMonTempRange
  * AnnMeanTemp
  * Isothemality  Mean Diurnal Range / Temperature Annual Range, with
    * Mean Diurnal Range  Mean of monthly (max temp - min temp)
    * Temperature Annual Range  Max Temperature of Warmest Month - Min Temperature of Coldest Month

**soil** is a 21 x 6 data.frame describing soil chemistery at each site. Each variable is reduced and centered

  * KLg  Logarithm of the potassium concentration
  * pH  Soil Ph
  * AlLg  Logarithm of the aluminium concentration
  * FeLg  Logarithm of the iron concentration
is_euclid

Test if the distance matrix is euclidean.

is_euclid(distances, tol = 1e-07)

Arguments

distances an object of class 'dist'
tol a tolerance threshold: an eigenvalue is considered positive if it is larger than -tol*lambda1 where lambda1 is the largest eigenvalue.

Author(s)

Christelle Gonindard-Melodelima
Eric Coissac

Examples

library(vegan)
data(bacteria)

bacteria_rel_freq <- sweep(bacteria, 1, rowSums(bacteria), "/")
bacteria_bray <- vegdist(bacteria_rel_freq, method = "bray")
is_euclid(bacteria_bray)

bacteria_chao <- vegdist(floor(bacteria*10000), method = "chao")
is_euclid(bacteria_chao)

---

**is_procmod_frame**

*Check if an object is a ProcMod Frame.*

**Description**

Check if an object is a ProcMod Frame.

**Usage**

```r
is_procmod_frame(x)
```

**Arguments**

- **x**
  - a R object to test

**Value**

A logical value equals to TRUE if `x` is a `procmod_frame`, FALSE otherwise.

**Author(s)**

Eric Coissac
Christelle Gonindard-Melodelima

**Examples**

```r
# Builds a procmod_frame with two random matrices
m1 <- simulate_matrix(10,20)
m2 <- simulate_matrix(10,30)
pmf <- procmod_frame(m1 = m1, m2 = m2)

# Returns TRUE
is_procmod_frame(pmf)

# Returns FALSE
is_procmod_frame(3)
```
The Names of the elements of a Correlation Matrix

Description

Returns the names of the elements associated to a procmod_corls object.

Usage

## S3 method for class 'procmod_corls'
names(x)

Arguments

x

a procmod_corls object

Author(s)

Eric Coissac

Christelle Gonindard-Melodelima

See Also

corls

Examples

# Build Three matrices of 3 rows.
A <- simulate_matrix(10,3)
B <- simulate_matrix(10,5)
C <- simulate_correlation(B,10,r2=0.6)
# Computes the correlation matrix
data <- procmod_frame(A = A, B = B, C = C)
cls <- corls(data, nrand = 100)

names(cls)
names.procmod_varls

The Names of the elements of a Variance / Covariance Matrix.

Description

Returns the names of the elements associated to a procmod_varls object.

Usage

```r
## S3 method for class 'procmod_varls'
names(x)
```

Arguments

- `x`: a procmod_varls object

Author(s)

Eric Coissac
Christelle Gonindard-Melodelima

See Also

- varls

Examples

```r
# Build Three matrices of 3 rows.
A <- simulate_matrix(10,3)
B <- simulate_matrix(10,5)
C <- simulate_correlation(B,10,r2=0.6)

# Computes the variance covariance matrix
data <- procmod_frame(A = A, B = B, C = C)
v <- varls(data, nrand = 100)

names(v)
```
Project a distance matrix in a euclidean space (NMDS).

Description

Project a set of points defined by a distance matrix in an euclidean space using the Kruskal’s Non-metric Multidimensional Scaling. This function is mainly a simplified interface on the `isoMDS` function using as much as possible dimensions to limit the stress. The aims of this NDMS being only to project point in an orthogonal space therefore without any correlation between axis. Because a non-metric method is used no condition is required on the used distance.

Usage

```r
nmds(distances, maxit = 100, trace = FALSE, tol = 0.001, p = 2)
```

Arguments

- `distances`: a `dist` object or a `matrix` object representing a distance matrix.
- `maxit`: The maximum number of iterations.
- `trace`: Logical for tracing optimization. Default `TRUE`.
- `tol`: Convergence tolerance.
- `p`: Power for Minkowski distance in the configuration space.

Value

A numeric matrix with at most \( n-1 \) dimensions, with \( n \) the number pf observations. This matrix defines the coordinates of each point in the orthogonal space.

Author(s)

Eric Coissac
Christelle Gonindard-Melodelima

Examples

```r
data(bacteria)
bacteria_rel_freq <- sweep(bacteria, 1, rowSums(bacteria), "/")
bacteria_hellinger <- sqrt(bacteria_rel_freq)
bacteria_dist <- dist(bacteria_hellinger)
project <- nmds(bacteria_dist)
```
Project a dataset in a euclidean space.

Description

Project a set of points defined by a distance matrix or a set of variables in an euclidean space. If the distance matrix is a metric, this is done using the `pcoa` function, for other distance the `nmds` is used. When points are described by a set of variable the `nmds` is used.

Usage

```r
ortho(data, ...) 
```

## S3 method for class 'dist'
```r
ortho(data, tol = 1e-07, ...)
```

## S3 method for class 'matrix'
```r
ortho(data, scale = FALSE, ...)
```

## S3 method for class 'data.frame'
```r
ortho(data, scale = FALSE, ...)
```

## S3 method for class 'procmod_frame'
```r
ortho(data, ...)
```

Arguments

- `data` a numeric matrix describing the points
- `...` other parameters specific to some implementation
- `tol` a tolerance threshold : an eigenvalue is considered positive if it is larger than `-tol*lambda1` where `lambda1` is the largest eigenvalue.
- `scale` a logical value indicating if the dimensions must be scaled to force for every column that `sd=1`. `FALSE` by default.

Value

A numeric matrix with at most n-1 dimensions, with n the number pf observations. This matrix defines the coordinates of each point in the orthogonal space.

Author(s)

Eric Coissac
Christelle Gonindard-Melodelima
Examples

```r
library(vegan)
data(bacteria)
data(eukaryotes)
data(soil)

dataset <- procmod_frame(euk = vegdist(decostand(eukaryotes,
                                  method = "hellinger"),
                                  method = "euclidean"),
                             bac = vegdist(decostand(bacteria,
                                  method = "hellinger"),
                                  method = "euclidean"),
                             soil = scale(soil,
                                  center = TRUE,
                                  scale = TRUE))

dp <- ortho(dataset)

bacteria_rel_freq <- sweep(bacteria,
                           1,
                           rowSums(bacteria),
                           "/")
bacteria_hellinger <- sqrt(bacteria_rel_freq)
bacteria_dist <- dist(bacteria_hellinger)

project <- ortho(bacteria_dist)
```

---

**pca**

*Project a set of points in a euclidean space (PCA).*

**Description**

Project a set of points defined by a set of numeric variables in an euclidean space using the principal component analysis. This function is mainly a simplified interface on the `prcomp` function using as much as possible dimensions to keep all the variation. The aims of this PCA being only to project point in an orthogonal space therefore without any correlation between axis. Data are centered by not scaled by default.

**Usage**

```r
pca(data, scale = FALSE)
```

**Arguments**

- `data` a numeric matrix describing the points
- `scale` a logical value indicating if the dimensions must be scaled to force for every column that sd=1. FALSE by default.
Value

a numeric matrix with at most \( n-1 \) dimensions, with \( n \) the number of observations. This matrix defines the coordinates of each point in the orthogonal space.

Author(s)

Eric Coissac
Christelle Gonindard-Melodelima

Examples

data(bacteria)
bacteria_rel_freq <- sweep(bacteria, 1, rowSums(bacteria), "/")
bacteria_hellinger <- sqrt(bacteria_rel_freq)
project <- pca(bacteria_hellinger)

pcoa

Project a distance matrix in a euclidean space (PCOA).

Description

Project a set of points defined by a distance matrix in an euclidean space using the Principal Co-
ordinates Analysis method. This function is mainly a simplified interface on the cmdscale function
using as much as possible dimensions for the projection. The aims of this PCoA being only to
project point in an orthogonal space therefore without any correlation between axis. Because a
metric method is used the used distance must be euclidean (cf is.euclid).

Usage

pcoa(distances)

Arguments

distances a dist object or a matrix object representing a distance matrix.

Value

a numeric matrix with at most \( n-1 \) dimensions, with \( n \) the number of observations. This matrix
defines the coordinates of each point in the orthogonal space.

Author(s)

Eric Coissac
Christelle Gonindard-Melodelima
print.procmod_corls

Examples

data(bacteria)
bacteria_rel_freq <- sweep(bacteria,
1,
rowSums(bacteria),
"
")
bacteria_hellinger <- sqrt(bacteria_rel_freq)
bacteria_dist <- dist(bacteria_hellinger)

project <- pcoa(bacteria_dist)

print.procmod_corls  

Print a procrustean Correlation Matrix.

Description

Print a procrustean Correlation Matrix.

Usage

## S3 method for class 'procmod_corls'
print(x, ...)

Arguments

x  
a procmod_corls object
...
other parameters passed to other functions

Author(s)

Eric Coissac
Christelle Gonindard-Melodelima

See Also

corls

Examples

# Build Three matrices of 3 rows.
A <- simulate_matrix(10,3)
B <- simulate_matrix(10,5)
C <- simulate_correlation(B,10,r2=0.6)

# Computes the correlation matrix
data <- procmad_frame(A = A, B = B, C = C)
cls <- corls(data, nrand = 100)
print.procmod_varls

Description
Print procrustean Variance / Covariance Matrix.

Usage
## S3 method for class 'procmod_varls'
print(x, ...)

Arguments
x a procmod_varls object
... other parameters passed to other functions

Author(s)
Eric Coissac
Christelle Gonindard-Melodelima

See Also
varls

Examples
# Build Three matrices of 3 rows.
A <- simulate_matrix(10,3)
B <- simulate_matrix(10,5)
C <- simulate_correlation(B,10, r2=0.6)

# Computes the variance covariance matrix
data <- procmod_frame(A = A, B = B, C = C)
v <- varls(data, nrand = 100)
print(v)
**procmod**

---

**Informative Procrustean Matrix Correlation**

---

**Description**

Estimates corrected Procrustean correlation between matrices for removing overfitting effect.

**Details**

The functions in the ProcMod package aims to estimate and to test correlation between matrices, correcting for the spurious correlations because of the over-fitting effect.

The ProcMod package is developed on the metabarcoding.org gitlab (https://git.metabarcoding.org/lecasofts/ProcMod). The gitlab of metabarcoding.org provides up-to-date information and forums for bug reports.

**Author(s)**

Christelle Gonindard-Melodelima

Eric Coissac

---

**procmod_frame**

*The procmod_frame data structure.*

---

**Description**

A procmod_frame can be considered as the analog of a data.frame for vector data. In a procmod_frame each element, equivalent to a column in a data.frame is a numeric matrix or a distance matrix object (dist). Every element must describe the same number of individuals. Therefore every numeric matrix must have the same number of row (nrow) and every distance matrix must have the same size (attr(d,"Size")). A procmod_frame can simultaneously contain both types of data, numeric and distance matrix.

**Usage**

```
procmod_frame(
  ..., 
  row_names = NULL, 
  check_rows = TRUE, 
  reorder_rows = TRUE, 
  contrasts_arg = NULL
)
```
Arguments

... a set of objects to aggregate into a procmod_frame. These objects can be numeric matrices, or dist objects. Every objects must have the same number of row.

row_names a character vector containing names associated to each row.

check_rows a logical value. When set to TRUE, its default value, the number of row of every elements of the procmod_frame are tested for equality. Otherwise no check is done.

reorder_rows a logical value. When set to TRUE, its default value, every elements of the procmod_frame are reordered according to the row_names order. Otherwise nothing is done.

contrasts_arg A list, whose entries are values (numeric matrices or character strings naming functions) to be used as replacement values for the contrasts replacement function and whose names are the names of columns of data containing factors.

Value

a procmod_frame instance.

Author(s)

Eric Coissac
Christelle Gonindard-Melodelima

Examples

library(vegan)
data(bacteria)
data(eukaryotes)
data(soil)

dataset <- procmod_frame(euk = vegdist(decostand(eukaryotes, method = "hellinger"),
            method = "euclidean"),
bac = vegdist(decostand(bacteria, method = "hellinger"),
            method = "euclidean"),
soil = scale(soil, center = TRUE,
            scale = TRUE))

length(dataset) nrow(dataset) ncol(dataset) dataset$euk
protate

Rotate the src matrix to fit into the space of the dest matrix.

Description

The optimal rotation is computed according to the procruste method. Rotation is based on singular value decomposition (SVD). No scaling and no centering are done, before computing the SVD.

Usage

protate(src, dest)

Arguments

src a numeric matrix to be rotated
dest a numeric matrix used as reference space

Value

a numeric matrix

Author(s)

Christelle Gonindard-Melodelima
Eric Coissac

Examples

# Generates two random matrices of size 10 x 15
m1 <- simulate_matrix(10, 15)
m2 <- simulate_matrix(10, 20)

# Rotates matrix m1 on m2
mr <- protate(m1, m2)

simulate_correlation

Simulate n points of dimension p correlated to a reference matrix.

Description

Simulates a set of points correlated to another set according to the procrustean correlation definition. Points are simulated by drawing values of each dimension from a normal distribution of mean 0 and standard deviation equals to 1. The mean of each dimension is forced to 0 (data are centred). By default variables are also scaled to enforce a standard deviation strictly equal to 1. Covariances between dimensions are not controlled. Therefore they are expected to be equal to 0 and reflect only the random distribution of the covariance between two random vectors. The intensity of the correlation is determined by the r2 parameter.
simulate_matrix

Usage

simulate_correlation(reference, p, r2, equal_var = TRUE)

Arguments

reference a numeric matrix to which the simulated data will be correlated
p an int value indicating the number of dimensions (variables) simulated
r2 the fraction of variation shared between the reference and the simulated data
equal_var a logical value indicating if the dimensions must be scaled to force sd=1. TRUE by default.

Value

a numeric matrix of nrow(reference) rows and p columns

Author(s)

Eric Coissac
Christelle Gonindard-Melodelima

Examples

sim1 <- simulate_matrix(25,10)
class(sim1)
dim(sim1)
sim2 <- simulate_correlation(sim1,20,0.8)
corls(sim1, sim2)^2

Description

Points are simulated by drawing values of each dimension from a normal distribution of mean 0 and standard deviation equals to 1. The mean of each dimension is forced to 0 (data are centred). By default variable are also scaled to enforce a standard deviation strictly equal to 1. Covariances between dimensions are not controled. Therefore they are expected to be equal to 0 and reflect only the random distribution of the covariance between two random vectors.

Usage

simulate_matrix(n, p, equal_var = TRUE)
Arguments

- **n**: an int value indicating the number of observations.
- **p**: an int value indicating the number of dimensions (variables) simulated.
- **equal_var**: a logical value indicating if the dimensions must be scaled to force sd=1. TRUE by default.

Value

a numeric matrix of n rows and p columns

Author(s)

Eric Coissac
Christelle Gonindard-Melodelima

Examples

```r
sim1 <- simulate_matrix(25,10)
class(sim1)
dim(sim1)
```

subset.procmod_frame  Subsetting Procmod Frames

Description

This is the implementation of the `subset` generic function for `procmod_frame`.

Usage

```r
## S3 method for class 'procmod_frame'
subset(x, subset, select, drop = FALSE, ...)
```

Arguments

- **x**: object to be subbed.
- **subset**: logical expression indicating elements or rows to keep: missing values are taken as false.
- **select**: expression, indicating columns to select from a data frame.
- **drop**: passed on to [ indexing operator.
- **...**: further arguments to be passed to or from other methods.
Details

The subset argument works on rows. Note that subset will be evaluated in the `procmod_frame`, so columns can be referred to (by name) as variables in the expression (see the examples).

The select argument if provided indicates with matrices have to be conserved. It works by first replacing column names in the selection expression with the corresponding column numbers in the `procmod_frame` and then using the resulting integer vector to index the columns. This allows the use of the standard indexing conventions so that for example ranges of columns can be specified easily, or single columns can be dropped (see the examples). Remember that each column of a `procmod_frame` is actually a matrix.

The drop argument is passed on to the `procmod_frame` indexing method. The default value is `FALSE`.

Value

A `procmod_frame` containing just the selected rows and columns.

Author(s)

Eric Coissac
Christelle Gonindard-Melodelima

Examples

```r
library(vegan)
data(bacteria)
data(eukaryotes)
data(soil)

dataset <- procmod_frame(euk = vegdist(decostand(eukaryotes, method = "hellinger"), method = "euclidean"),
bac = vegdist(decostand(bacteria, method = "hellinger"), method = "euclidean"),
soil = scale(soil, center = TRUE, scale = TRUE))

dim(dataset)

higher_ph = subset(dataset,soil[,"pH"] > 0)
dim(higher_ph)

without_bacteria = subset(dataset,soil[,"pH"] > 0, -bac)
dim(without_bacteria)
```
varls, corls compute the procrustean variance / covariance, or correlation matrices between a set of real matrices and dist objects.

Usage

```r
varls(..., nrand = 100, p_adjust_method = "holm")
corls(..., nrand = 100, p_adjust_method = "holm")
```

Arguments

- `...` the set of matrices or a procmod_frame object.
- `nrand` number of randomisation used to estimate the mean covariance observed between two random matrix. If rand is NULL or equal to 0, no correction is estimated and the raw procrustean covariances are estimated.
- `p_adjust_method` the multiple test correction method used to adjust p values. p_adjust_method belongs one of the following values: "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none". The default is set to "holm".

Details

Procrustean covariance between two matrices X and Y, is defined as the sum of the singular values of the X'Y matrix (Gower 1971; Lingoes and Schönemann 1974). Both the X and Y matrices must have the same number of rows.

The variances and covariances and correlations are corrected to avoid over fitting (Coissac and Gonindard-Melodelima 2019).

The inputs must be numeric matrices or dist object. The set of input matrices can be aggregated un a procmod_frame.

Before computing the coefficients, matrices are projected into an orthogonal space using the ortho function.

The denominator n - 1 is used which gives an unbiased estimator of the (co)variance for i.i.d. observations.

Value

a procmod_varls object which corresponds to a numeric matrix annotated by several attributes.

The following attribute is always added:

- `nrand` an integer value indicating the number of randomisations used to estimate the mean of the random covariance.
When `nrand` is greater than 0 a couple of attributes is added:

- `rcovls` a numeric matrix containing the estimation of the mean of the random covariance.
- `p.value` a numeric matrix containing the estimations of the p.values of tests checking that the observed covariance is larger than the mean of the random covariance. p.values are corrected for multiple tests according to the method specified by the `p.adjust_method` parameter.

**Author(s)**

Eric Coissac
Christelle Gonindard-Melodelima

**References**


**See Also**

`p.adjust`

**Examples**

```r
# Build Three matrices of 3 rows.
A <- simulate_matrix(10,3)
B <- simulate_matrix(10,5)
C <- simulate_correlation(B,10,r2=0.6)

# Computes the variance covariance matrix
varls(A = A, B = B, C = C)

data = procmod_frame(A = A, B = B, C = C)
varls(data)

# Computes the correlation matrix
corls(data, nrand = 100)
```
Index

* datasets
  eukaryotes, 9
  .Trace, 4
  .getPermuteMatrix, 2
  .procmod_coerce_value, 3
  .rep_matrix, 3

  as.data.frame.dist, 5
  as.procmod_frame, 6

  bacteria(eukaryotes), 9
  bicenter, 7

  climat(eukaryotes), 9
  cmdscale, 18
  corls, 13, 19
  corls(varls), 27
  corls_test, 8

  dim.procmod_frame, 9
  dist, 15, 18, 27

  eukaryotes, 9

  geography(eukaryotes), 9

  how, 2, 8

  is.euclid, 11
  is.euclid, 11, 18
  is.procmod_frame, 12
  isoMDS, 15

  matrix, 5, 15, 18

  names.procmod_corls, 13
  names.procmod_varls, 14
  nmds, 15, 16
  nobs, 2
  ortho, 16, 27

  p.adjust, 8, 28
  pca, 17
  pcoa, 16, 18
  prcomp, 17
  print.procmod_corls, 19
  print.procmod_varls, 20
  procmod, 21
  procmod_frame, 3, 8, 9, 21, 27
  procuste.rtest, 8
  protate, 23

  simulate_correlation, 23
  simulate_matrix, 24
  soil(eukaryotes), 9
  subset, 25
  subset.procmod_frame, 25

  varls, 14, 20, 27