Package 'dimensio'

April 9, 2024

Title Multivariate Data Analysis

Version 0.7.0

Maintainer Nicolas Frerebeau <`nicolas.frerebeau@u-bordeaux-montaigne.fr`>

Description Simple Principal Components Analysis (PCA) and (Multiple)
Correspondence Analysis (CA) based on the Singular Value Decomposition
(SVD). This package provides S4 classes and methods to compute,
extract, summarize and visualize results of multivariate data
analysis. It also includes methods for partial bootstrap validation
described in Greenacre (1984, ISBN: 978-0-12-299050-2) and Lebart et
al. (2006, ISBN: 978-2-10-049616-7).

License GPL (>= 3)

URL <https://packages.tesselle.org/dimensio/>,

<https://github.com/tesselle/dimensio>

BugReports <https://github.com/tesselle/dimensio/issues>

Depends R (>= 3.5)

Imports arkhe (>= 1.6.0), graphics, grDevices, methods

Suggests khroma, knitr, markdown, rsvg, svglite, tinysnapshot,
tinytest

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Collate 'AllClasses.R' 'AllGenerics.R' 'augment.R' 'biplot.R'
'bootstrap.R' 'ca.R' 'coerce.R' 'data.R' 'dimensio-defunct.R'
'dimensio-deprecated.R' 'dimensio-internal.R'
'dimensio-package.R' 'get_contributions.R' 'get_coordinates.R'
'get_correlations.R' 'get_cos2.R' 'get_data.R'
'get_distances.R' 'get_eigenvalues.R' 'get_inertia.R'
'get_variance.R' 'loadings.R' 'mca.R' 'mutators.R' 'pca.R'
'predicates.R' 'predict.R' 'reexport.R' 'screeplot.R' 'show.R'
'subset.R' 'summary.R' 'svd.R' 'tidy.R' 'tools.R'

'viz_contributions.R' 'viz_coordinates.R' 'viz_cos2.R'
'viz_ellipse.R' 'viz_hull.R' 'viz_labels.R' 'wrap_ellipses.R'
'wrap_hull.R' 'zzz.R'

NeedsCompilation no

Author Nicolas Frerebeau [aut, cre] (<https://orcid.org/0000-0001-5759-4944>,
    Université Bordeaux Montaigne),
    Jean-Baptiste Fourvel [ctb] (<https://orcid.org/0000-0002-1061-4642>,
    CNRS),
    Brice Lebrun [ctb] (<https://orcid.org/0000-0001-7503-8685>, Logo
    designer),
    Université Bordeaux Montaigne [fnd],
    CNRS [fnd]

R **topics documented:**

---

benthos	*Benthos*

Description

Abundances of Marine Species in Sea-Bed Samples

Usage

```
benthos
```

Format

A [data.frame](#) with 13 columns (sites) and 92 rows (species).

Source

<http://www.carme-n.org/?sec=data7>

See Also

Other datasets: [colours](#), [countries](#)

---

biplot	*Biplot*

Description

Biplot

Usage

```
## S4 method for signature 'CA'
biplot(
  x,
  ...,
  axes = c(1, 2),
  type = c("symetric", "rows", "columns", "contributions"),
  active = TRUE,
  sup = TRUE,
  labels = NULL,
  col.rows = c("#E69F00", "#009E73"),
  col.columns = c("#56B4E9", "#F0E442"),
  cex.rows = graphics::par("cex"),
  cex.columns = graphics::par("cex"),
  pch.rows = 16,
```

```
    pch.columns = 17,
    xlim = NULL,
    ylim = NULL,
    main = NULL,
    sub = NULL,
    legend = list(x = "topleft")
  )

  ## S4 method for signature 'PCA'
  biplot(
    x,
    ...,
    axes = c(1, 2),
    type = c("form", "covariance"),
    active = TRUE,
    sup = TRUE,
    labels = "variables",
    col.rows = c("#E69F00", "#009E73"),
    col.columns = c("#56B4E9", "#F0E442"),
    pch = 16,
    cex = 1,
    lty = 1,
    lwd = 2,
    xlim = NULL,
    ylim = NULL,
    main = NULL,
    sub = NULL,
    legend = list(x = "topleft")
  )
```

Arguments

x	A [CA](), [MCA]() or [PCA]() object.
...	Currently not used.
axes	A length-two [numeric]() vector giving the dimensions to be plotted.
type	A [character]() string specifying the biplot to be plotted (see below). It must be one of "rows", "columns", "contribution" (CA), "form" or "covariance" (PCA). Any unambiguous substring can be given.
active	A [logical]() scalar: should the active observations be plotted?
sup	A [logical]() scalar: should the supplementary observations be plotted?
labels	A [character]() vector specifying whether "rows"/"individuals" and/or "columns"/"variables" names must be drawn. Any unambiguous substring can be given.
col.rows	A length-two vector of color specification for the active and supplementary rows.
col.columns	A length-two vector of color specification for the active and supplementary columns.

xlim	A length-two [numeric](#) vector giving the x limits of the plot. The default value, NULL, indicates that the range of the [finite](#) values to be plotted should be used.
ylim	A length-two [numeric](#) vector giving the y limits of the plot. The default value, NULL, indicates that the range of the [finite](#) values to be plotted should be used.
main	A [character](#) string giving a main title for the plot.
sub	A [character](#) string giving a subtitle for the plot.
legend	A [list](#) of additional arguments to be passed to [graphics::legend()](#); names of the list are used as argument names. If NULL, no legend is displayed.
pch, pch.rows, pch.columns	
	A symbol specification.
cex, cex.rows, cex.columns	
	A [numeric](#) vector giving the amount by which plotting characters and symbols should be scaled relative to the default.
lty, lwd	A specification for the line type and width.

Details

A biplot is the simultaneous representation of rows and columns of a rectangular dataset. It is the generalization of a scatterplot to the case of mutlivariate data: it allows to visualize as much information as possible in a single graph (Greenacre 2010).

Biplots have the drawbacks of their advantages: they can quickly become difficult to read as they display a lot of information at once. It may then be preferable to visualize the results for individuals and variables separately.

Value

biplot() is called for its side-effects: it results in a graphic being displayed. Invisibly returns x.

PCA Biplots

form (**row-metric-preserving**) The form biplot favors the representation of the individuals: the distance between the individuals approximates the Euclidean distance between rows. In the form biplot the length of a vector approximates the quality of the representation of the variable.

covariance (**column-metric-preserving**) The covariance biplot favors the representation of the variables: the length of a vector approximates the standard deviation of the variable and the cosine of the angle formed by two vectors approximates the correlation between the two variables. In the covariance biplot the distance between the individuals approximates the Mahalanobis distance between rows.

CA Biplots

symetric (**symetric biplot**) Represents the row and column profiles simultaneously in a common space: rows and columns are in standard coordinates. Note that the the inter-distance between any row and column items is not meaningful.

rows (**asymetric biplot**) Row principal biplot (row-metric-preserving) with rows in principal coordinates and columns in standard coordinates.

columns **(asymetric biplot)** Column principal biplot (column-metric-preserving) with rows in standard coordinates and columns in principal coordinates.

contribution **(asymetric biplot)** Contribution biplot with rows in principal coordinates and columns in standard coordinates multiplied by the square roots of their masses.

Author(s)

N. Frerebeau

References

Aitchison, J. and Greenacre, M. J. (2002). Biplots of Compositional Data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 51(4): 375-92. doi:10.1111/14679876.00275.

Greenacre, M. J. (2010). *Biplots in Practice*. Bilbao: Fundación BBVA.

See Also

Other plot methods: `screeplot()`, `viz_contributions()`, `viz_individuals()`, `viz_variables()`, `viz_wrap`, `wrap`

Examples

```
## Replicate examples from Greenacre 2007, p. 59-68
data("countries")

## Compute principal components analysis
## All rows and all columns obtain the same weight
row_w <- rep(1 / nrow(countries), nrow(countries)) # 1/13
col_w <- rep(1 / ncol(countries), ncol(countries)) # 1/6
Y <- pca(countries, scale = FALSE, weight_row = row_w, weight_col = col_w)

## Row-metric-preserving biplot (form biplot)
biplot(Y, type = "form")

## Column-metric-preserving biplot (covariance biplot)
biplot(Y, type = "covariance", legend = list(x = "bottomright"))

## Replicate examples from Greenacre 2007, p. 79-88
data("benthos")

## Compute correspondence analysis
X <- ca(benthos)

## Symetric CA biplot
biplot(X, labels = "columns", legend = list(x = "bottomright"))

## Row principal CA biplot
biplot(X, type = "row", labels = "columns", legend = list(x = "bottomright"))

## Column principal CA biplot
biplot(X, type = "column", labels = "columns", legend = list(x = "bottomright"))
```

```
## Contribution CA biplot
biplot(X, type = "contrib", labels = NULL, legend = list(x = "bottomright"))
```

---

boot   *Partial Bootstrap Analysis*

Description

Checks analysis with partial bootstrap resampling.

Usage

```
## S4 method for signature 'CA'
bootstrap(object, n = 30)

## S4 method for signature 'PCA'
bootstrap(object, n = 30)
```

Arguments

object	A CA or PCA object.
n	A non-negative integer giving the number of bootstrap replications.

Value

Returns a BootstrapCA or a BootstrapPCA object.

Author(s)

N. Frerebeau

References

Greenacre, Michael J. *Theory and Applications of Correspondence Analysis*. London: Academic Press, 1984.

Lebart, L., Piron, M. and Morineau, A. *Statistique exploratoire multidimensionnelle: visualisation et inférence en fouille de données*. Paris: Dunod, 2006.

Lockyear, K. (2013). Applying Bootstrapped Correspondence Analysis to Archaeological Data. *Journal of Archaeological Science*, 40(12): 4744-4753. doi:10.1016/j.jas.2012.08.035.

Ringrose, T. J. (1992). Bootstrapping and Correspondence Analysis in Archaeology. *Journal of Archaeological Science*, 19(6): 615-629. doi:10.1016/03054403(92)90032X.

Examples

```
## Bootstrap on CA
## Data from Lebart et al. 2006, p. 170-172
data("colours")

## Compute correspondence analysis
X <- ca(colours)

## Bootstrap (30 replicates)
Y <- bootstrap(X, n = 30)

## Not run:
## Get replicated coordinates
get_replications(Y, margin = 1)
get_replications(Y, margin = 2)

## End(Not run)

## Plot with ellipses
viz_rows(Y)
viz_tolerance(Y, margin = 1, level = c(0.68, 0.95))

viz_columns(Y)
viz_tolerance(Y, margin = 2, level = c(0.68, 0.95))

## Plot with convex hulls
viz_columns(Y)
viz_hull(Y, margin = 2)

## Bootstrap on PCA
## Compute principal components analysis
data("iris")
X <- pca(iris)

## Bootstrap (30 replicates)
Y <- bootstrap(X, n = 30)

## Plot with ellipses
viz_variables(Y)
viz_tolerance(Y, margin = 2, level = c(0.68, 0.95))
```

---

burt   *Burt Table*

Description

Computes the burt table of a factor table.

Usage

```
burt(object, ...)

## S4 method for signature 'data.frame'
burt(object, exclude = NULL, abbrev = TRUE)
```

Arguments

object	A [data.frame]().
...	Currently not used.
exclude	A vector of values to be excluded when forming the set of levels (see [factor()]()). If NULL (the default), will make NA an extra level.
abbrev	A [logical]() scalar: should the column names be abbreviated? If FALSE, these are of the form 'factor_level' but if abbrev = TRUE they are just 'level' which will suffice if the factors have distinct levels.

Value

A symetric [matrix]().

Author(s)

N. Frerebeau

See Also

Other tools: [cdt]()()

Examples

```
## Create a factor table
x <- data.frame(
  A = c("a", "b", "a"),
  B = c("x", "y", "z")
)

## Complete disjunctive table
cdt(x)

## Burt table
burt(x)
```

---

ca   *Correspondence Analysis*

Description

Computes a simple correspondence analysis based on the singular value decomposition.

Usage

```
ca(object, ...)

## S4 method for signature 'data.frame'
ca(object, rank = NULL, sup_row = NULL, sup_col = NULL)

## S4 method for signature 'matrix'
ca(object, rank = NULL, sup_row = NULL, sup_col = NULL)
```

Arguments

object	A $m \times p$ numeric matrix or a data.frame.
...	Currently not used.
rank	An integer value specifying the maximal number of components to be kept in the results. If NULL (the default), $min(m, p) - 1$ components will be returned.
sup_row	A vector specifying the indices of the supplementary rows.
sup_col	A vector specifying the indices of the supplementary columns.

Value

A CA object.

Author(s)

N. Frerebeau

References

Greenacre, M. J. *Theory and Applications of Correspondence Analysis*. London: Academic Press, 1984.

Greenacre, M. J. *Correspondence Analysis in Practice*. Seconde edition. Interdisciplinary Statistics Series. Boca Raton: Chapman & Hall/CRC, 2007.

Lebart, L., Piron, M. and Morineau, A. *Statistique exploratoire multidimensionnelle: visualisation et inférence en fouille de données*. Paris: Dunod, 2006.

See Also

svd()

Other multivariate analysis: mca(), pca(), predict()

Examples

```
## Data from Lebart et al. 2006, p. 170-172
data("colours")

## The chi square of independence between the two variables
stats::chisq.test(colours)

## Compute correspondence analysis
X <- ca(colours)

## Plot rows
viz_rows(X, labels = TRUE)

## Plot columns
viz_columns(X, labels = TRUE)

## Get row coordinates
head(get_coordinates(X, margin = 1))

## Get column coordinates
head(get_coordinates(X, margin = 2))

## Get row distances to centroid
head(get_distances(X, margin = 1))

## Get row inertias
head(get_inertia(X, margin = 1))

## Get row contributions
head(get_contributions(X, margin = 1))

## Get eigenvalues
get_eigenvalues(X)
```

---

cdt   *Complete Disjunctive Table*

Description

Computes the complete disjunctive table of a factor table.

Usage

```
cdt(object, ...)

## S4 method for signature 'matrix'
cdt(object, exclude = NULL, abbrev = TRUE)

## S4 method for signature 'data.frame'
cdt(object, exclude = NULL, abbrev = TRUE)
```

Arguments

object	A `data.frame`.
...	Currently not used.
exclude	A `vector` of values to be excluded when forming the set of levels (see `factor()`). If `NULL` (the default), will make `NA` an extra level.
abbrev	A `logical` scalar: should the column names be abbreviated? If `FALSE`, these are of the form 'factor_level' but if `abbrev = TRUE` they are just 'level' which will suffice if the factors have distinct levels.

Value

A `data.frame`.

Author(s)

N. Frerebeau

See Also

Other tools: `burt()`

Examples

```
## Create a factor table
x <- data.frame(
  A = c("a", "b", "a"),
  B = c("x", "y", "z")
)

## Complete disjunctive table
cdt(x)

## Burt table
burt(x)
```

---

colours	*Colours*

Description

Contingency table of eye and hair colours of different individuals.

Usage

```
colours
```

Format

A [data.frame](#) with 4 columns (hair colours) and 4 rows (eye colours).

Source

Lebart, L., Piron, M. and Morineau, A. *Statistique exploratoire multidimensionnelle: visualisation et inférence en fouille de données*. Paris: Dunod, 2006, p. 170-172

See Also

Other datasets: [benthos](#), [countries](#)

---

countries *Countries*

Description

Student ratings of 13 countries on six attributes.

Usage

countries

Format

A [data.frame](#) with 6 columns (attributes) and 13 rows (countries).

Source

Greenacre, M. J. *Biplots in Practice*. Bilbao: Fundación BBVA, 2010.

See Also

Other datasets: [benthos](#), [colours](#)

---

dimnames   *Dimnames of an Object*

Description

Retrieve or set the dimnames of an object.

Usage

```
## S4 method for signature 'MultivariateAnalysis'
dim(x)

## S4 method for signature 'MultivariateAnalysis'
rownames(x, do.NULL = TRUE, prefix = "row")

## S4 method for signature 'MultivariateAnalysis'
colnames(x, do.NULL = TRUE, prefix = "col")

## S4 method for signature 'MultivariateAnalysis'
dimnames(x)
```

Arguments

x	An object from which to retrieve the row or column names (a CA or PCA object).
do.NULL	A logical scalar. If FALSE and names are NULL, names are created.
prefix	A character string specifying the prefix for created names.

Author(s)

N. Frerebeau

See Also

Other mutators: get_contributions(), get_coordinates(), get_data(), get_distances(),
get_eigenvalues(), loadings(), subset()

---

get_contributions   *Get Contributions*

Description

Get Contributions

Usage

```
get_contributions(x, ...)

get_correlations(x, ...)

get_cos2(x, ...)

## S4 method for signature 'MultivariateAnalysis'
get_contributions(x, margin = 1)

## S4 method for signature 'PCA'
get_correlations(x, sup_name = ".sup")

## S4 method for signature 'MultivariateAnalysis'
get_cos2(x, margin = 1, sup_name = ".sup")
```

Arguments

x	An object from which to get element(s) (a CA, MCA or PCA object).
...	Currently not used.
margin	A length-one numeric vector giving the subscript which the data will be returned: 1 indicates individuals/rows (the default), 2 indicates variables/columns.
sup_name	A character string specifying the name of the column to create for supplementary points attribution (see below).

Value

- `get_contributions()` returns a data.frame of contributions to the definition of the principal dimensions.
- `get_correlations()` returns a data.frame of correlations between variables and dimensions (PCA). An extra column (named after sup_name) is added specifying whether an observation is a supplementary point or not.
- `get_cos2()` returns a data.frame of cos^2 values (i.e. quality of the representation of the points on the factor map). An extra column (named after sup_name) is added specifying whether an observation is a supplementary point or not.

Author(s)

N. Frerebeau

See Also

Other mutators: dimnames(), get_coordinates(), get_data(), get_distances(), get_eigenvalues(), loadings(), subset()

get_coordinates   *Get Coordinates*

Description

Get Coordinates

Usage

```
get_coordinates(x, ...)

get_replications(x, ...)

## S4 method for signature 'MultivariateAnalysis'
get_coordinates(x, margin = 1, principal = TRUE, sup_name = ".sup")

## S4 method for signature 'MultivariateBootstrap'
get_replications(x, margin = 1)

## S4 method for signature 'BootstrapPCA'
get_replications(x)
```

Arguments

x	An object from which to get element(s) (a CA, MCA or PCA object).
...	Currently not used.
margin	A length-one numeric vector giving the subscript which the data will be returned: 1 indicates individuals/rows (the default), 2 indicates variables/columns.
principal	A logical scalar: should principal coordinates be returned? If FALSE, standard coordinates are returned.
sup_name	A character string specifying the name of the column to create for supplementary points attribution (see below).

Value

- get_coordinates() returns a data.frame of coordinates. An extra column (named after sup_name) is added specifying whether an observation is a supplementary point or not.
- get_replications() returns an array of coordinates.

Author(s)

N. Frerebeau

See Also

Other mutators: dimnames(), get_contributions(), get_data(), get_distances(), get_eigenvalues(), loadings(), subset()

Examples

```
## Load data
data("iris")

## Compute principal components analysis
X <- pca(iris, scale = TRUE, sup_row = 5:10)

## Get row principal coordinates
head(get_coordinates(X, margin = 1, principal = TRUE))

## Get row standard coordinates
head(get_coordinates(X, margin = 1, principal = FALSE))

## Tidy principal coordinates
head(tidy(X, margin = 1))
head(tidy(X, margin = 2))

head(augment(X, margin = 1, axes = c(1, 2)))
head(augment(X, margin = 2, axes = c(1, 2)))
```

---

get_data   *Get Original Data*

Description

Get Original Data

Usage

```
get_data(x, ...)

## S4 method for signature 'MultivariateAnalysis'
get_data(x)
```

Arguments

x   An object from which to get element(s) (a CA, MCA or PCA object).

...   Currently not used.

Value

Returns a data.frame of original data.

Author(s)

N. Frerebeau

See Also

Other mutators: dimnames(), get_contributions(), get_coordinates(), get_distances(), get_eigenvalues(), loadings(), subset()

---

get_distances   *Get Distances*

Description

Get Distances

Usage

```
get_distances(x, ...)

## S4 method for signature 'MultivariateAnalysis'
get_distances(x, margin = 1)
```

Arguments

x	An object from which to get element(s) (a CA, MCA or PCA object).
...	Currently not used.
margin	A length-one numeric vector giving the subscript which the data will be returned: 1 indicates individuals/rows (the default), 2 indicates variables/columns.

Value

A numeric vector of squared distance to the centroide.

Author(s)

N. Frerebeau

See Also

Other mutators: dimnames(), get_contributions(), get_coordinates(), get_data(), get_eigenvalues(), loadings(), subset()

---

get_eigenvalues *Get Eigenvalues*

Description

Get Eigenvalues

Usage

```
get_eigenvalues(x)

get_inertia(x, ...)

get_variance(x, ...)

## S4 method for signature 'MultivariateAnalysis'
get_eigenvalues(x)

## S4 method for signature 'MultivariateAnalysis'
get_inertia(x, margin = 1)

## S4 method for signature 'MultivariateAnalysis'
get_variance(x, digits = 2)
```

Arguments

x	An object from which to get element(s) (a CA, MCA or PCA object).
...	Currently not used.
margin	A length-one numeric vector giving the subscript which the data will be returned: 1 indicates individuals/rows (the default), 2 indicates variables/columns.
digits	An integer indicating the number of decimal places to be used.

Value

- get_eigenvalues() returns a data.frame with the following columns: eigenvalues, variance (percentage of variance) and cumulative (cumulative percentage of variance).
- get_variance() returns a numeric vector giving the percentage of explained variance of each dimension.
- get_inertia() returns a numeric vector.

Author(s)

N. Frerebeau

See Also

Other mutators: dimnames(), get_contributions(), get_coordinates(), get_data(), get_distances(),
loadings(), subset()

---

loadings   *Extract Loadings*

Description

Extract loadingsin principal components analysis.

Usage

```
## S4 method for signature 'PCA'
loadings(x, ...)
```

Arguments

x   A PCA object.

...   Currently not used.

Value

Returns variable loadings (i.e. the coefficients of the linear combination of the original variables).

Note

loadings() is only implemented for consistency with stats::loadings().

Author(s)

N. Frerebeau

See Also

Other mutators: dimnames(), get_contributions(), get_coordinates(), get_data(), get_distances(),
get_eigenvalues(), subset()

---

mca | *Multiple Correspondence Analysis*

Description

Computes a multiple correspondence analysis.

Usage

```
mca(object, ...)

## S4 method for signature 'data.frame'
mca(object, rank = NULL, sup_row = NULL, sup_col = NULL, sup_quanti = NULL)

## S4 method for signature 'matrix'
mca(object, rank = NULL, sup_row = NULL, sup_col = NULL)
```

Arguments

object	A $m \times p$ numeric matrix or a data.frame.
...	Currently not used.
rank	An integer value specifying the maximal number of components to be kept in the results. If NULL (the default), $min(m, p) - 1$ components will be returned.
sup_row	A vector specifying the indices of the supplementary rows.
sup_col	A vector specifying the indices of the supplementary categorical columns.
sup_quanti	A vector specifying the indices of the supplementary quantitative columns.

Value

A MCA object.

Author(s)

N. Frerebeau

References

Lebart, L., Piron, M. and Morineau, A. *Statistique exploratoire multidimensionnelle: visualisation et inférence en fouille de données*. Paris: Dunod, 2006.

See Also

svd(), cdt()

Other multivariate analysis: ca(), pca(), predict()

---

pca | *Principal Components Analysis*

Description

Computes a principal components analysis based on the singular value decomposition.

Usage

```
pca(object, ...)

## S4 method for signature 'data.frame'
pca(
  object,
  center = TRUE,
  scale = TRUE,
  rank = NULL,
  sup_row = NULL,
  sup_col = NULL,
  sup_quali = NULL,
  weight_row = NULL,
  weight_col = NULL
)

## S4 method for signature 'matrix'
pca(
  object,
  center = TRUE,
  scale = TRUE,
  rank = NULL,
  sup_row = NULL,
  sup_col = NULL,
  weight_row = NULL,
  weight_col = NULL
)
```

Arguments

object	A $m \times p$ numeric [matrix](#) or a [data.frame](#).
...	Currently not used.
center	A [logical](#) scalar: should the variables be shifted to be zero centered?
scale	A [logical](#) scalar: should the variables be scaled to unit variance?
rank	An [integer](#) value specifying the maximal number of components to be kept in the results. If NULL (the default), $p - 1$ components will be returned.
sup_row	A vector specifying the indices of the supplementary rows.

sup_col	A `vector` specifying the indices of the supplementary columns.
sup_quali	A `vector` specifying the indices of the supplementary qualitative columns.
weight_row	A [numeric](#) vector specifying the active row (individual) weights. If NULL (the default), uniform weights are used. Row weights are internally normalized to sum 1
weight_col	A [numeric](#) vector specifying the active column (variable) weights. If NULL (the default), uniform weights (1) are used.

Value

A [PCA](#) object.

Author(s)

N. Frerebeau

References

Lebart, L., Piron, M. and Morineau, A. *Statistique exploratoire multidimensionnelle: visualisation et inférence en fouille de données*. Paris: Dunod, 2006.

See Also

[svd()](#)

Other multivariate analysis: [ca](#)(), [mca](#)(), [predict](#)()

Examples

```
## Load data
data("iris")

## Compute principal components analysis
X <- pca(iris, scale = TRUE, sup_row = sample(150, 10), sup_col = 2)

## Get row coordinates
head(get_coordinates(X, margin = 1))

## Get column coordinates
head(get_coordinates(X, margin = 2))

## Get correlations between variables and dimensions
head(get_correlations(X))

## Get eigenvalues
get_eigenvalues(X)
```

---

predict   *Predict New Coordinates*

Description

Predict the projection of new individuals/rows or variables/columns.

Usage

```
## S4 method for signature 'CA'
predict(object, newdata, margin = 1)

## S4 method for signature 'MCA'
predict(object, newdata, margin = 1)

## S4 method for signature 'PCA'
predict(object, newdata, margin = 1)
```

Arguments

object	A CA or PCA object.
newdata	An object of supplementary points coercible to a matrix for which to compute principal coordinates.
margin	A length-one numeric vector giving the subscript which the data will be predicted: 1 indicates individuals/rows (the default), 2 indicates variables/columns.

Value

A data.frame of coordinates.

Author(s)

N. Frerebeau

See Also

Other multivariate analysis: ca(), mca(), pca()

Examples

```
## Create a matrix
A <- matrix(data = sample(1:10, 100, TRUE), nrow = 10, ncol = 10)

## Compute correspondence analysis
X <- ca(A, sup_row = 8:10, sup_col = 7:10)

## Predict new row coordinates
Y <- matrix(data = sample(1:10, 120, TRUE), nrow = 20, ncol = 6)
```

```
predict(X, Y, margin = 1)

## Predict new column coordinates
Z <- matrix(data = sample(1:10, 140, TRUE), nrow = 7, ncol = 20)
predict(X, Z, margin = 2)
```

---

screeplot	*Scree Plot*

Description

Plot eigenvalues (scree plot) or variances histogram.

Usage

```
## S4 method for signature 'MultivariateAnalysis'
screeplot(
  x,
  ...,
  eigenvalues = FALSE,
  cumulative = FALSE,
  labels = TRUE,
  limit = 10,
  col = "grey90",
  border = "grey10",
  col.cumulative = "red",
  lty.cumulative = "solid",
  lwd.cumulative = 2
)
```

Arguments

x	A CA, MCA or PCA object.
...	Extra parameters to be passed to graphics::barplot().
eigenvalues	A logical scalar: should the eigenvalues be plotted instead of variance/inertia?
cumulative	A logical scalar: should the cumulative percentages of variance be plotted?
labels	A logical scalar: should text labels be drawn on top of bars?
limit	An integer specifying the number of top elements to be displayed.
col, border	A character string specifying the bars infilling and border colors.
col.cumulative	A specification for the line color.
lty.cumulative	A specification for the line type.
lwd.cumulative	A specification for the line width.

Value

screeplot() is called for its side-effects: it results in a graphic being displayed. Invisibly returns x.

Author(s)

N. Frerebeau

See Also

Other plot methods: biplot(), viz_contributions(), viz_individuals(), viz_variables(), viz_wrap, wrap

Examples

```
## Load data
data("iris")

## Compute principal components analysis
X <- pca(iris, scale = TRUE)

## Screeplot
screeplot(X)
screeplot(X, cumulative = TRUE)
```

---

subset   *Extract Parts of an Object*

Description

Operators acting on objects to extract parts.

Usage

```
## S4 method for signature 'CA,ANY,missing'
x[[i]]

## S4 method for signature 'PCA,ANY,missing'
x[[i]]
```

Arguments

x   An object from which to extract element(s) or in which to replace element(s).

i   A character string specifying elements to extract. Any unambiguous substring can be given (see details).

Details

If i is "data", returns a list with the following elements:

data A [numeric](#) matrix of raw data.

mean A [numeric](#) vector giving the variables means (PCA).

sd A [numeric](#) vector giving the variables standard deviations (PCA).

If i is "rows", returns a list with the following elements:

coord A [numeric](#) matrix of rows/individuals coordinates.

cos2 A [numeric](#) matrix of rows/individuals squared cosine.

masses A [numeric](#) vector giving the rows masses/individual weights.

sup A [logical](#) vector specifying whether a point is a supplementary observation or not.

If i is "columns", returns a list with the following elements:

coord A [numeric](#) matrix of columns/variables coordinates.

cor A [numeric](#) matrix of correlation between variables and the dimensions (PCA).

cos2 A [numeric](#) matrix of columns/variables squared cosine.

masses A [numeric](#) vector giving the columns masses/variable weights.

sup A [logical](#) vector specifying whether a point is a supplementary observation or not.

If i is "eigenvalues", returns a [numeric](#) vector of eigenvalues.

Value

A [list](#).

Author(s)

N. Frerebeau

See Also

Other mutators: [dimnames()](#), [get_contributions()](#), [get_coordinates()](#), [get_data()](#), [get_distances()](#), [get_eigenvalues()](#), [loadings()](#)

Examples

```
## Load data
data("iris")

## Compute principal components analysis
X <- pca(iris, scale = TRUE, sup_row = 8:10, sup_col = 1)

## Get results for the individuals
X[["rows"]]
```

---

summary   *Object Summaries*

Description

Provides a summary of the results of a multivariate data analysis.

Usage

```
## S4 method for signature 'CA'
summary(object, ..., margin = 1, active = TRUE, sup = TRUE, rank = 3)

## S4 method for signature 'PCA'
summary(object, ..., margin = 1, active = TRUE, sup = TRUE, rank = 3)
```

Arguments

object	A CA, MCA or PCA object.
...	Currently not used.
margin	A length-one numeric vector giving the subscript which the data will be summarized: 1 indicates individuals/rows (the default), 2 indicates variables/columns.
active	A logical scalar: should the active observations be summarized?
sup	A logical scalar: should the supplementary observations be summarized?
rank	An integer value specifying the maximal number of components to be kept in the results.

Author(s)

N. Frerebeau

See Also

Other summary: tidy()

Examples

```
## Data from Lebart et al. 2006, p. 170-172
data("colours")

## Compute correspondence analysis
X <- ca(colours)

## Rows summary
summary(X, margin = 1)

## Columns summary
summary(X, margin = 2)
```

---

tidy   *Tidy Coordinates*

Description

Tidy Coordinates

Usage

```
tidy(x, ...)

augment(x, ...)

## S4 method for signature 'MultivariateAnalysis'
augment(x, ..., margin = 1, axes = c(1, 2), principal = TRUE)

## S4 method for signature 'MultivariateAnalysis'
tidy(x, ..., margin = 1, principal = TRUE)
```

Arguments

x	A [CA](), [MCA]() or [PCA]() object.
...	Currently not used.
margin	A length-one [numeric]() vector giving the subscript which the data will be returned: 1 indicates individuals/rows (the default), 2 indicates variables/columns.
axes	A length-two [numeric]() vector giving the dimensions for which to compute results.
principal	A [logical]() scalar: should principal coordinates be returned? If FALSE, standard coordinates are returned.

Value

tidy() returns a long [data.frame]() with the following columns:

label Row/column names of the original data.

component Component.

supplementary Whether an observation is active or supplementary.

coordinate Coordinates.

contribution Contributions to the definition of the components.

cos2 cos^2.

augment() returns a wide [data.frame]() of the row/column coordinates along axes and the following columns:

label Row/column names of the original data.

supplementary Whether an observation is active or supplementary.

mass Weight/mass of each observation.

sum Sum of squared coordinates along axes.

contribution Joint contributions to the definition of axes.

cos2 Joint cos^2 along axes.

Author(s)

N. Frerebeau

See Also

Other summary: [summary](}()

Examples

```
## Load data
data("iris")

## Compute principal components analysis
X <- pca(iris, scale = TRUE, sup_row = 5:10)

## Get row principal coordinates
head(get_coordinates(X, margin = 1, principal = TRUE))

## Get row standard coordinates
head(get_coordinates(X, margin = 1, principal = FALSE))

## Tidy principal coordinates
head(tidy(X, margin = 1))
head(tidy(X, margin = 2))

head(augment(X, margin = 1, axes = c(1, 2)))
head(augment(X, margin = 2, axes = c(1, 2)))
```

---

viz_contributions   *Visualize Contributions and cos2*

Description

Plots contributions histogram and cos^2 scatterplot.

Usage

```
viz_contributions(x, ...)

viz_cos2(x, ...)

## S4 method for signature 'MultivariateAnalysis'
viz_contributions(
  x,
  ...,
  margin = 2,
  axes = 1,
  sort = TRUE,
  decreasing = TRUE,
  limit = 10,
  horiz = FALSE,
  col = "grey90",
  border = "grey10"
)

## S4 method for signature 'MultivariateAnalysis'
viz_cos2(
  x,
  ...,
  margin = 2,
  axes = c(1, 2),
  active = TRUE,
  sup = TRUE,
  sort = TRUE,
  decreasing = TRUE,
  limit = 10,
  horiz = FALSE,
  col = "grey90",
  border = "grey10"
)
```

Arguments

x	A CA, MCA or PCA object.
...	Extra parameters to be passed to graphics::barplot().
margin	A length-one numeric vector giving the subscript which the data will be returned: 1 indicates individuals/rows (the default), 2 indicates variables/columns.
axes	A numeric vector giving the dimensions to be plotted.
sort	A logical scalar: should the data be sorted?
decreasing	A logical scalar: should the sort order be decreasing? Only used if sort is TRUE.
limit	An integer specifying the number of top elements to be displayed.

horiz	A [logical](#) scalar: should the bars be drawn horizontally with the first at the bottom?
col, border	A [character](#) string specifying the bars infilling and border colors.
active	A [logical](#) scalar: should the active observations be plotted?
sup	A [logical](#) scalar: should the supplementary observations be plotted?

Details

The red dashed line indicates the expected average contribution (variables with a contribution larger than this cutoff can be considered as important in contributing to the component).

Value

viz_contributions() and viz_cos2() are called for their side-effects: they result in a graphic being displayed. Invisibly return x.

Author(s)

N. Frerebeau

See Also

Other plot methods: [biplot](#)(), [screeplot](#)(), [viz_individuals](#)(), [viz_variables](#)(), [viz_wrap](#), [wrap](#)

Examples

```
## Load data
data("iris")

## Compute principal components analysis
X <- pca(iris, scale = TRUE)

## Get row contributions
head(get_contributions(X, margin = 1))

## Plot contributions
viz_contributions(X, axes = 1)

## Plot cos2
viz_cos2(X)
```

viz_individuals   *Visualize Individuals Factor Map*

Description

Plots row/individual principal coordinates.

Usage

```
viz_individuals(x, ...)

viz_rows(x, ...)

## S4 method for signature 'MultivariateAnalysis'
viz_rows(
  x,
  ...,
  axes = c(1, 2),
  active = TRUE,
  sup = TRUE,
  labels = FALSE,
  labels_max = 10,
  highlight = NULL,
  xlim = NULL,
  ylim = NULL,
  main = NULL,
  sub = NULL,
  panel.first = NULL,
  panel.last = NULL,
  legend = list(x = "topleft")
)

## S4 method for signature 'BootstrapCA'
viz_rows(x, ..., axes = c(1, 2))

## S4 method for signature 'PCA'
viz_individuals(
  x,
  ...,
  axes = c(1, 2),
  active = TRUE,
  sup = TRUE,
  labels = FALSE,
  labels_max = 10,
  highlight = NULL,
  xlim = NULL,
  ylim = NULL,
```

```
    main = NULL,
    sub = NULL,
    panel.first = NULL,
    panel.last = NULL,
    legend = list(x = "topleft")
)
```

Arguments

x	A CA, MCA or PCA object.
...	Further graphical parameters (see details).
axes	A length-two numeric vector giving the dimensions to be plotted.
active	A logical scalar: should the active observations be plotted?
sup	A logical scalar: should the supplementary observations be plotted?
labels	A logical scalar: should labels be drawn?
labels_max	An integer specifying the number of labels to draw. Only the labels of the n observations with highest cos^2 values will be drawn. If NULL, all labels are drawn. Only used if labels is TRUE.
highlight	A vector specifying the information to be highlighted. If NULL (the default), no highlighting is applied. If a single character string is passed, it must be the name of a categorical variable, or one of "observation", "mass", "sum", "contribution" or "cos2" (see augment()).
xlim	A length-two numeric vector giving the x limits of the plot. The default value, NULL, indicates that the range of the finite values to be plotted should be used.
ylim	A length-two numeric vector giving the y limits of the plot. The default value, NULL, indicates that the range of the finite values to be plotted should be used.
main	A character string giving a main title for the plot.
sub	A character string giving a subtitle for the plot.
panel.first	An expression to be evaluated after the plot axes are set up but before any plotting takes place. This can be useful for drawing background grids.
panel.last	An expression to be evaluated after plotting has taken place but before the axes, title and box are added.
legend	A list of additional arguments to be passed to graphics::legend(); names of the list are used as argument names. If NULL, no legend is displayed.

Details

Commonly used graphical parameters are:

pch A vector of plotting characters or symbols. This can either be a single character or an integer code for one of a set of graphics symbols.

cex A numerical vector giving the amount by which plotting characters and symbols should be scaled relative to the default.

col The colors for lines and points. Multiple colors can be specified so that each point can be given its own color.

bg The background color for the open plot symbols given by pch = 21:25.

Value

viz_*() is called for its side-effects: it results in a graphic being displayed. Invisibly returns x.

Note

Be careful: graphical parameters are silently recycled.

Author(s)

N. Frerebeau

See Also

Other plot methods: biplot(), screeplot(), viz_contributions(), viz_variables(), viz_wrap, wrap

Examples

```
## Load data
data("iris")

## Compute principal components analysis
X <- pca(iris, scale = TRUE)

## Plot individuals
viz_individuals(X, panel.last = graphics::grid())

## Plot variables
viz_variables(X, panel.last = graphics::grid())

## Graphical parameters
## Continuous values
viz_individuals(X, highlight = iris$Petal.Length, pch = 16)
viz_individuals(X, highlight = iris$Petal.Length, pch = 16,
                col = grDevices::hcl.colors(12, "RdPu"))
viz_individuals(X, highlight = iris$Petal.Length, pch = 16,
                col = grDevices::hcl.colors(12, "RdPu"),
                cex = c(1, 2))

viz_variables(X, highlight = "contribution",
              col = grDevices::hcl.colors(12, "BluGrn", rev = TRUE),
              lwd = c(1, 5))

## Discrete values
viz_individuals(X, highlight = iris$Species, pch = 21:23)
viz_individuals(X, highlight = iris$Species, pch = 21:23,
                bg = c("#004488", "#DDAA33", "#BB5566"),
                col = "black")

viz_variables(X, highlight = c("Petal", "Petal", "Sepal", "Sepal"),
              col = c("#EE7733", "#0077BB"),
              lty = c(1, 3))
```

---

viz_variables   *Visualize Variables Factor Map*

Description

Plots column/variable principal coordinates.

Usage

```
viz_variables(x, ...)

viz_columns(x, ...)

## S4 method for signature 'MultivariateAnalysis'
viz_columns(
  x,
  ...,
  axes = c(1, 2),
  active = TRUE,
  sup = TRUE,
  labels = FALSE,
  labels_max = 10,
  highlight = NULL,
  xlim = NULL,
  ylim = NULL,
  main = NULL,
  sub = NULL,
  panel.first = NULL,
  panel.last = NULL,
  legend = list(x = "topleft")
)

## S4 method for signature 'MultivariateBootstrap'
viz_columns(x, ..., axes = c(1, 2))

## S4 method for signature 'PCA'
viz_variables(
  x,
  ...,
  axes = c(1, 2),
  active = TRUE,
  sup = TRUE,
  labels = TRUE,
  labels_max = 10,
  highlight = NULL,
  xlim = NULL,
  ylim = NULL,
```

```
  main = NULL,
  sub = NULL,
  panel.first = NULL,
  panel.last = NULL,
  legend = list(x = "topleft")
)

## S4 method for signature 'CA'
viz_variables(
  x,
  ...,
  axes = c(1, 2),
  active = TRUE,
  sup = TRUE,
  labels = FALSE,
  labels_max = 10,
  highlight = NULL,
  xlim = NULL,
  ylim = NULL,
  main = NULL,
  sub = NULL,
  panel.first = NULL,
  panel.last = NULL,
  legend = list(x = "topleft")
)

## S4 method for signature 'BootstrapPCA'
viz_variables(x, ..., axes = c(1, 2))
```

Arguments

x	A [CA](), [MCA]() or [PCA]() object.
...	Further [graphical parameters]() (see details).
axes	A length-two [numeric]() vector giving the dimensions to be plotted.
active	A [logical]() scalar: should the active observations be plotted?
sup	A [logical]() scalar: should the supplementary observations be plotted?
labels	A [logical]() scalar: should labels be drawn?
labels_max	An [integer]() specifying the number of labels to draw. Only the labels of the n variables contributing the most to the factorial map will be drawn. If NULL, all labels are drawn. Only used if labels is TRUE.
highlight	A vector specifying the information to be highlighted. If NULL (the default), no highlighting is applied. If a single character string is passed, it must be the name of a categorical variable, or one of "observation", "mass", "sum", "contribution" or "cos2" (see [augment()]()).
xlim	A length-two [numeric]() vector giving the x limits of the plot. The default value, NULL, indicates that the range of the [finite]() values to be plotted should be used.

ylim	A length-two [numeric](#) vector giving the y limits of the plot. The default value, NULL, indicates that the range of the [finite](#) values to be plotted should be used.
main	A [character](#) string giving a main title for the plot.
sub	A [character](#) string giving a subtitle for the plot.
panel.first	An expression to be evaluated after the plot axes are set up but before any plotting takes place. This can be useful for drawing background grids.
panel.last	An expression to be evaluated after plotting has taken place but before the axes, title and box are added.
legend	A [list](#) of additional arguments to be passed to [graphics::legend()](#); names of the list are used as argument names. If NULL, no legend is displayed.

Details

Commonly used [graphical parameters](#) are:

pch   A vector of plotting characters or symbols. This can either be a single character or an integer code for one of a set of graphics symbols.

cex   A numerical vector giving the amount by which plotting characters and symbols should be scaled relative to the default.

lty   A vector of line types.

lwd   A vector of line widths.

col   The colors for lines and points. Multiple colors can be specified so that each point can be given its own color.

bg   The background color for the open plot symbols given by pch = 21:25.

Value

viz_*() is called for its side-effects: it results in a graphic being displayed. Invisibly returns x.

Note

Be careful: graphical parameters are silently recycled.

Author(s)

N. Frerebeau

See Also

Other plot methods: [biplot()](#), [screeplot()](#), [viz_contributions()](#), [viz_individuals()](#), [viz_wrap](#), [wrap](#)

Examples

```
## Load data
data("iris")

## Compute principal components analysis
X <- pca(iris, scale = TRUE)

## Plot individuals
viz_individuals(X, panel.last = graphics::grid())

## Plot variables
viz_variables(X, panel.last = graphics::grid())

## Graphical parameters
## Continuous values
viz_individuals(X, highlight = iris$Petal.Length, pch = 16)
viz_individuals(X, highlight = iris$Petal.Length, pch = 16,
                col = grDevices::hcl.colors(12, "RdPu"))
viz_individuals(X, highlight = iris$Petal.Length, pch = 16,
                col = grDevices::hcl.colors(12, "RdPu"),
                cex = c(1, 2))

viz_variables(X, highlight = "contribution",
              col = grDevices::hcl.colors(12, "BluGrn", rev = TRUE),
              lwd = c(1, 5))

## Discrete values
viz_individuals(X, highlight = iris$Species, pch = 21:23)
viz_individuals(X, highlight = iris$Species, pch = 21:23,
                bg = c("#004488", "#DDAA33", "#BB5566"),
                col = "black")

viz_variables(X, highlight = c("Petal", "Petal", "Sepal", "Sepal"),
              col = c("#EE7733", "#0077BB"),
              lty = c(1, 3))
```

---

viz_wrap   *Plot Envelopes*

Description

Plot Envelopes

Usage

```
viz_hull(x, ...)

viz_confidence(x, ...)
```

```
viz_tolerance(x, ...)

## S4 method for signature 'MultivariateAnalysis'
viz_tolerance(x, ..., margin = 1, axes = c(1, 2), group = NULL, level = 0.95)

## S4 method for signature 'BootstrapCA'
viz_tolerance(x, ..., margin = 1, axes = c(1, 2), level = 0.95)

## S4 method for signature 'MultivariateAnalysis'
viz_confidence(x, ..., margin = 1, axes = c(1, 2), group = NULL, level = 0.95)

## S4 method for signature 'BootstrapCA'
viz_confidence(x, ..., margin = 1, axes = c(1, 2), level = 0.95)

## S4 method for signature 'MultivariateAnalysis'
viz_hull(x, ..., margin = 1, axes = c(1, 2), group = NULL)

## S4 method for signature 'BootstrapCA'
viz_hull(x, ..., margin = 1, axes = c(1, 2))
```

Arguments

x	An object from which to wrap observations (a CA, MCA or PCA object).
...	Further graphical parameters to be passed to graphics::polygon().
margin	A length-one numeric vector giving the subscript which the data will be returned: 1 indicates individuals/rows (the default), 2 indicates variables/columns.
axes	A length-two numeric vector giving the dimensions for which to compute results.
group	A vector specifying the group an observation belongs to, or a single character string giving the name of a categorical variable.
level	A numeric vector specifying the confidence/tolerance level.

Value

viz_*()is called for its side-effects: it results in a graphic being displayed. Invisibly returns x.

Author(s)

N. Frerebeau

See Also

Other plot methods: biplot(), screeplot(), viz_contributions(), viz_individuals(), viz_variables(), wrap

Examples

```
## Load data
data("iris")

## Compute principal components analysis
X <- pca(iris, scale = TRUE, sup_quali = "Species")

## Plot with convex hulls
col <- c("#004488", "#DDAA33", "#BB5566")
viz_rows(X, highlight = "Species", col = col)
viz_hull(X, group = "Species", border = col)

## Plot with tolerance ellipses
col <- c("#004488", "#DDAA33", "#BB5566")
viz_rows(X, highlight = "Species", col = col)
viz_tolerance(X, group = "Species", border = col)
```

---

wrap   *Wrap Observations*

Description

- wrap_hull() computes convex hull of a set of observations.
- wrap_confidence() computes a confidence ellipse.
- wrap_tolerance() computes a tolerance ellipse.

Usage

```
wrap_hull(x, ...)

wrap_confidence(x, ...)

wrap_tolerance(x, ...)

## S4 method for signature 'MultivariateAnalysis'
wrap_confidence(x, margin = 1, axes = c(1, 2), group = NULL, level = 0.95)

## S4 method for signature 'MultivariateAnalysis'
wrap_tolerance(x, margin = 1, axes = c(1, 2), group = NULL, level = 0.95)

## S4 method for signature 'MultivariateAnalysis'
wrap_hull(x, margin = 1, axes = c(1, 2), group = NULL)
```

Arguments

x   An object from which to wrap observations (a CA, MCA or PCA object).
...   Currently not used.

margin	A length-one [numeric](#) vector giving the subscript which the data will be returned: 1 indicates individuals/rows (the default), 2 indicates variables/columns.
axes	A length-two [numeric](#) vector giving the dimensions for which to compute results.
group	A vector specifying the group an observation belongs to, or a single `character` string giving the name of a categorical variable.
level	A [numeric](#) vector specifying the confidence/tolerance level.

Value

`wrap_*()` returns a [data.frame](#) of envelope x and y coordinates.

An extra column named `group` is added specifying the group an observation belongs to.

Author(s)

N. Frerebeau

See Also

Other plot methods: [biplot()](#), [screeplot()](#), [viz_contributions()](#), [viz_individuals()](#), [viz_variables()](#), [viz_wrap](#)

Examples

```
## Load data
data("iris")

## Compute principal components analysis
X <- pca(iris, scale = TRUE, sup_quali = "Species")

## Confidence ellipse coordinates
conf <- wrap_confidence(X, margin = 1, group = "Species", level = c(0.68, 0.95))

## Tolerance ellipse coordinates
conf <- wrap_confidence(X, margin = 1, group = "Species", level = 0.95)

## Convex hull coordinates
hulls <- wrap_hull(X, margin = 1, group = "Species")
```

Index