Package ‘dimensio’

October 13, 2022

Title Multivariate Data Analysis

Version 0.3.0

Maintainer Nicolas Frerebeau

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

Description Simple Principal Components Analysis (PCA) and 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 (>= 3.3)

Imports ggplot2, grDevices, methods, rlang

Suggests covr, FactoMineR, ggrepel, khroma, testthat (>= 3.0.0), vdiffr (>= 1.0.0)

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.2.1

Collate 'AllClasses.R' 'AllGenerics.R' 'bootstrap.R' 'ca.R' 'data.R' 'dimensio-package.R' 'ggplot2.R' 'helpers.R' 'mutators.R' 'pca.R' 'plot.R' 'show.R' 'subset.R' 'summary.R' 'svd.R' 'tidy.R' 'wrap.R' 'zzz.R'

LazyData true

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>, Université Bordeaux Montaigne)

Repository CRAN

Date/Publication 2022-08-15 14:50:05 UTC

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("rows", "columns", "contributions"),
  active = TRUE,
  sup = TRUE,
  label = c("rows", "columns")
)

## S4 method for signature 'PCA'
biplot(
  x,
  axes = c(1, 2),
  type = c("form", "covariance"),
  active = TRUE,
  sup = TRUE,
  label = c("individuals", "variables")
)
```

Arguments

`x` A `CA` or `PCA` object.

`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?

`label` A `character` vector specifying whether "`rows`"/"`individuals`" and/or "`columns`"/"`variables`" names must be mapped (e.g. for use with `ggrepel::geom_label_repel()`). Any unambiguous substring can be given.

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

A ggplot2::ggplot object.

PCA Biplots

`form` Form biplot (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` Covariance biplot (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

`rows` Row principal biplot.

`columns` Column principal biplot.

`contribution` Contribution biplot.

Author(s)

N. Frerebeau

References

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

See Also

Other plot methods: `envelopes`, `plot_contributions()`, `plot_coordinates`, `plot_eigenvalues`

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") +
  ggrepel::geom_label_repel()

## Column-metric-preserving biplot (covariance biplot)
biplot(Y, type = "covariance") +
  ggrepel::geom_label_repel()

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

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

## Row principal CA biplot
biplot(X, type = "row") +
  ggrepel::geom_label_repel()

## Column principal CA biplot
biplot(X, type = "column") +
  ggrepel::geom_label_repel()

## Contribution CA biplot
biplot(X, type = "contrib") +
  ggrepel::geom_label_repel()
```

bootstrap *Partial Bootstrap Analysis*

Description

Checks analysis with partial bootstrap resampling.

Usage

```
bootstrap(object, ...)

## 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.
...	Currently not used.
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.

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
plot_rows(Y, colour = ”group”) +
  ggplot2::stat_ellipse()

## Plot with convex hulls
plot_columns(Y, colour = ”group”, fill = ”group”) +
  stat_hull(geom = ”polygon”, alpha = 0.5)

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

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

## Plot with ellipses
plot_columns(Y, colour = ”group”) +
  ggplot2::stat_ellipse()
```

`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 numeric or logical vector specifying the indices of the supplementary rows.
sup_col	A numeric or logical 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: 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)
```

```
## 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)
```

CA-class *CA Results*

Description

An S4 class to store the results of a simple correspondence analysis.

Author(s)

N. Frerebeau

See Also

Other class: MultivariateAnalysis, MultivariateBootstrap, MultivariateResults, MultivariateSummary, PCA-class

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)

## 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)
```

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`

envelopes *Wrap Observations*

Description

- `wrap_hull()` computes convex hull of a set of observations.

Usage

```
wrap_hull(x, ...)

stat_hull(
  mapping = NULL,
  data = NULL,
  geom = "polygon",
  position = "identity",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  ...
)

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

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

Arguments

`x` An object from which to wrap observations (a `CA` or `PCA` object).

`...` Currently not used.

`mapping` Set of aesthetic mappings created by `aes()` or `aes_()`. If specified and `inherit.aes = TRUE` (the default), it is combined with the default mapping at the top level of the plot. You must supply `mapping` if there is no plot mapping.

`data` The data to be displayed in this layer. There are three options:

If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot()`.

A `data.frame`, or other object, will override the plot data. All objects will be fortified to produce a data frame. See `fortify()` for which variables will be created.

A `function` will be called with a single argument, the plot data. The return value must be a `data.frame`, and will be used as the layer data. A `function` can be created from a `formula` (e.g. `~ head(.x, 10)`).

`geom` The geometric object to use display the data

`position` Position adjustment, either as a string, or the result of a call to a position adjustment function.

`na.rm` A `logical` scalar: should missing values be silently removed? If `FALSE` (the), missing values are removed with a warning.

`show.legend` logical. Should this layer be included in the legends? `NA`, the default, includes if any aesthetics are mapped. `FALSE` never includes, and `TRUE` always includes. It can also be a named logical vector to finely select the aesthetics to display.

`inherit.aes` If `FALSE`, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. `borders()`.

`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 to be for which to compute results.

`group` A vector specifying the group an observation belongs to.

Value

- `stat_hull()` return a `ggplot2::layer()`.
- `wrap_hull()` return a `data.frame` of envelope principal coordinates. An extra column named `group` is added specifying the group an observation belongs to.

Author(s)

N. Frerebeau

References

https://ggplot2.tidyverse.org/articles/extending-ggplot2.html

See Also

Other plot methods: `biplot()`, `plot_contributions()`, `plot_coordinates`, `plot_eigenvalues`

Examples

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

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

## Plot results
plot_rows(X, colour = "group", group = iris$Species) +
    khroma::scale_colour_highcontrast()

## Convex hull coordinates
hulls <- wrap_hull(X, group = iris$Species)
head(hulls)

## Plot with convex hulls
plot_rows(X, colour = "group", group = iris$Species) +
  stat_hull(geom = "path") +
  khroma::scale_colour_highcontrast()
```

`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` 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: `get_coordinates()`, `get_data()`, `get_distances()`, `get_eigenvalues()`, `mutators`, `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 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: get_contributions(), get_data(), get_distances(), get_eigenvalues(), mutators, 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` or `PCA` object).
`...`	Currently not used.

Value

Returns a `data.frame` of original data.

Author(s)

N. Frerebeau

See Also

Other mutators: `get_contributions()`, `get_coordinates()`, `get_distances()`, `get_eigenvalues()`, `mutators`, `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` 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

- `get_distances()` returns a `numeric` vector of squared distance to the centroide.

Author(s)

N. Frerebeau

See Also

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

`MultivariateAnalysis` *Output of Multivariate Data Analysis*

Description

A virtual S4 class to store the output of a multivariate data analysis.

Slots

`data` A `numeric` `matrix`.

`dimension` An `integer` giving the dimension of the solution.

`singular_values` A `numeric` vector giving the singular values.

`rows` A `MultivariateResults` object.

`columns` A `MultivariateResults` object.

Subset

In the code snippets below, `x` is a `MultivariateAnalysis` object.

`x[[i]]` Extracts information from a slot selected by subscript `i`. `i` is a length-one `character` vector.

Author(s)

N. Frerebeau

See Also

Other class: CA-class, MultivariateBootstrap, MultivariateResults, MultivariateSummary, PCA-class

---

MultivariateBootstrap *Output of Bootstrap Replications*

Description

A virtual S4 class to store the output of a bootstrap analysis.

Slots

`replications` An integer giving the number of bootstrap replications.

Author(s)

N. Frerebeau

See Also

Other class: CA-class, MultivariateAnalysis, MultivariateResults, MultivariateSummary, PCA-class

---

MultivariateResults *Multivariate Data Analysis Results*

Description

An S4 class to store the results of a multivariate data analysis.

Slots

`names` A character vector specifying the row names.

`principal` A numeric matrix giving the principal coordinates.

`standard` A numeric matrix giving the standard coordinates.

`contributions` A numeric matrix giving the contributions to the definition of the dimensions.

`cosine` A numeric matrix giving the cos^2 values.

`distances` A numeric vector giving the distances to centroid.

`weights` A numeric vector giving the masses/weights.

- `supplement` A `logical` vector specifying the supplementary points.
- `order` An `integer` vector giving the original indices of the data (computation moves all supplementary points at the end of the results).
- `groups` A `character` vector specifying the class for each observation.

Author(s)

N. Frerebeau

See Also

Other class: `CA-class`, `MultivariateAnalysis`, `MultivariateBootstrap`, `MultivariateSummary`, `PCA-class`

`MultivariateSummary` *Summary of Multivariate Data Analysis*

Description

A virtual S4 class to store the summary of a multivariate data analysis.

Slots

- `data` A `numeric` `matrix`.
- `eigenvalues` A `numeric` `matrix`.
- `results` A `numeric` `matrix`.
- `supplement` A `logical` vector specifying the supplementary points.
- `margin` An `integer`.

Author(s)

N. Frerebeau

See Also

Other class: `CA-class`, `MultivariateAnalysis`, `MultivariateBootstrap`, `MultivariateResults`, `PCA-class`

mutators *Get Results*

Description

Getters to retrieve parts of an object.

Usage

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

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

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

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

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

Arguments

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

Value

`loadings()` returns variable loadings (i.e. the coefficients of the linear combination of the original variables). `loadings()` is only implemented for consistency with **[stats][stats::loadings]**.

Author(s)

N. Frerebeau

See Also

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

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,
  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 `numeric` or `logical` vector specifying the indices of the supplementary rows (individuals).

`sup_col` A `numeric` or `logical` vector specifying the indices of the supplementary columns (variables).

`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()`, `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 row contributions
head(get_contributions(X, margin = 1))

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

## Get eigenvalues
get_eigenvalues(X)
```

PCA-class *PCA Results*

Description

An S4 class to store the results of a principal components analysis.

Slots

center A numeric vector giving the column mean of the initial dataset (active individuals only).

scale A numeric vector giving the column standard deviations of the initial dataset (active individuals only).

Author(s)

N. Frerebeau

See Also

Other class: CA-class, MultivariateAnalysis, MultivariateBootstrap, MultivariateResults, MultivariateSummary

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

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

## Get eigenvalues
get_eigenvalues(X)
```

plot_contributions *Visualize Contributions and cos2*

Description

Plots contributions histogram and cos^2 scatterplot.

Usage

```
plot_contributions(object, ...)

plot_cos2(object, ...)

## S4 method for signature 'MultivariateAnalysis'
plot_contributions(
  object,
  margin = 2,
  axes = 1,
  sort = TRUE,
  decreasing = TRUE,
  limit = 10,
  fill = "grey30",
  border = "grey10"
)

## S4 method for signature 'MultivariateAnalysis'
plot_cos2(
  object,
  margin = 2,
  axes = c(1, 2),
  active = TRUE,
  sup = TRUE,
  sort = TRUE,
  decreasing = TRUE,
  limit = 10,
  fill = "grey30",
  border = "grey10"
)
```

Arguments

`object`	A `CA` 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-one `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.

`fill, 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?

Value

A ggplot2::ggplot object.

Author(s)

N. Frerebeau

See Also

Other plot methods: `biplot()`, `envelopes`, `plot_coordinates`, `plot_eigenvalues`

Examples

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

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

## Plot results
plot_individuals(Y, colour = "group", shape = "group", group = iris$Species) +
  khroma::scale_colour_highcontrast()

plot_individuals(Y, colour = "group", size = "cos2", group = iris$Sepal.Width) +
  khroma::scale_color_YlOrBr()

plot_individuals(Y, colour = "contribution", size = "contribution") +
  khroma::scale_color_iridescent(range = c(0.5, 1))

plot_variables(Y, colour = "contribution") +
  ggrepel::geom_label_repel() +
  khroma::scale_color_YlOrBr(range = c(0.5, 1))
```

plot_coordinates *Visualize Factor Map*

Description

Plots principal coordinates.

Usage

```
plot_rows(object, ...)

plot_columns(object, ...)

plot_individuals(object, ...)

plot_variables(object, ...)

## S4 method for signature 'MultivariateAnalysis'
plot_rows(
  object,
  axes = c(1, 2),
  active = TRUE,
  sup = TRUE,
  alpha = NULL,
  colour = NULL,
  fill = NULL,
  shape = NULL,
  size = NULL,
  group = NULL
)

## S4 method for signature 'MultivariateAnalysis'
plot_columns(
  object,
  axes = c(1, 2),
  active = TRUE,
  sup = TRUE,
  alpha = NULL,
  colour = NULL,
  fill = NULL,
  shape = NULL,
  size = NULL,
  group = NULL
)

## S4 method for signature 'BootstrapPCA'
plot_columns(
  object,
  axes = c(1, 2),
  active = TRUE,
  sup = TRUE,
  alpha = NULL,
  colour = NULL,
  fill = NULL,
  shape = NULL,
  size = NULL,
  group = NULL
)

## S4 method for signature 'PCA'
plot_individuals(
  object,
  axes = c(1, 2),
  active = TRUE,
  sup = TRUE,
  alpha = NULL,
  colour = NULL,
  fill = NULL,
  shape = NULL,
  size = NULL,
  group = NULL
)

## S4 method for signature 'PCA'
plot_variables(
  object,
  axes = c(1, 2),
  active = TRUE,
  sup = TRUE,
  alpha = NULL,
  colour = NULL,
  linetype = NULL,
  size = NULL,
  group = NULL
)
```

Arguments

`object` A `CA` or `PCA` object.

`...` Currently not used.

`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?

`alpha, colour, fill, linetype, shape, size`
A `character` string specifying the information to be highlighted (will be mapped

to the corresponding aesthetic). It must be one of "observation", "mass", "sum", "contribution", "cos2" or "group" (see details). Any unambiguous substring can be given. If NULL (the default), no highlighting is applied.

group
A vector of categories specifying the categorical variable from which to highlight the individuals (only used if at least one of colour, fill, linetype or shape is set to group; see details).

Details

observation 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.

Value

A ggplot2::ggplot object.

Author(s)

N. Frerebeau

See Also

Other plot methods: biplot(), envelopes, plot_contributions(), plot_eigenvalues

Examples

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

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

## Plot results
plot_individuals(Y, colour = "group", shape = "group", group = iris$Species) +
  khroma::scale_colour_highcontrast()

plot_individuals(Y, colour = "group", size = "cos2", group = iris$Sepal.Width) +
  khroma::scale_color_YlOrBr()

plot_individuals(Y, colour = "contribution", size = "contribution") +
  khroma::scale_color_iridescent(range = c(0.5, 1))

plot_variables(Y, colour = "contribution") +
  ggrepel::geom_label_repel() +
  khroma::scale_color_YlOrBr(range = c(0.5, 1))
```

plot_eigenvalues *Visualize Eigenvalues*

Description

Plot eigenvalues or variances histogram.

Usage

```
plot_variance(object, ...)

## S4 method for signature 'MultivariateAnalysis'
plot_variance(
  object,
  variance = TRUE,
  cumulative = TRUE,
  fill = "grey30",
  border = "grey10",
  colour = "red"
)
```

Arguments

`object`	A `CA` or `PCA` object.
`...`	Currently not used.
`variance`	A `logical` scalar: should the percentages of variance be plotted instead of the eigenvalues?
`cumulative`	A `logical` scalar: should the cumulative percentages of variance be plotted?
`fill, border`	A `character` string specifying the bars infilling and border colors.
`colour`	A `character` string specifying the line color.

Value

A ggplot2::ggplot object.

Author(s)

N. Frerebeau

See Also

Other plot methods: `biplot()`, `envelopes`, `plot_contributions()`, `plot_coordinates`

Examples

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

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

## Plot results
plot_individuals(Y, colour = "group", shape = "group", group = iris$Species) +
  khroma::scale_colour_highcontrast()

plot_individuals(Y, colour = "group", size = "cos2", group = iris$Sepal.Width) +
  khroma::scale_color_YlOrBr()

plot_individuals(Y, colour = "contribution", size = "contribution") +
  khroma::scale_color_iridescent(range = c(0.5, 1))

plot_variables(Y, colour = "contribution") +
  ggrepel::geom_label_repel() +
  khroma::scale_color_YlOrBr(range = c(0.5, 1))
```

---

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 '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()`, `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)
```

`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: get_contributions(), get_coordinates(), get_data(), get_distances(), get_eigenvalues(), mutators

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` or `PCA` object.
`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

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'
tidy(x, margin = 1, principal = TRUE, ...)

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

Arguments

`x`	A `CA` 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.
`axes`	A length-two `numeric` vector giving the dimensions to be for which to compute results.

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

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)))
```

Index