

Package ‘dimensio’

June 22, 2026

Title Multivariate Data Analysis

Version 0.14.2

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://codeberg.org/tesselle/dimensio>,
<https://packages.tesselle.org/dimensio/>,
<https://tesselle.r-universe.dev/dimensio>

BugReports <https://codeberg.org/tesselle/dimensio/issues>

Depends R (>= 3.5)

Imports arkhe (>= 1.11.0), graphics, grDevices, khroma (>= 1.17.0),
methods, utils

Suggests fontquiver, knitr, markdown, rsvg, svglite, tinysnapshot (>= 0.2.0), tinytest

VignetteBuilder knitr

Encoding UTF-8

LazyData true

X-schema.org-isPartOf <https://www.tesselle.org>

X-schema.org-keywords data-analysis, multivariate-analysis, r-package

Collate 'AllClasses.R' 'AllGenerics.R' 'annotations.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' 'export.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' 'mca.R' 'mutators.R' 'pca.R'
'pcoa.R' 'plot.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' 'wrap_ellipses.R' 'wrap_hull.R'
'zzz.R'
```

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Nicolas Frerebeau [aut, cre] (ORCID:

<<https://orcid.org/0000-0001-5759-4944>>),

Jean-Baptiste Fourvel [ctb] (ORCID:

<<https://orcid.org/0000-0002-1061-4642>>),

Camille Thabard [ctb] (ORCID: <<https://orcid.org/0000-0002-3196-7658>>),

Brice Lebrun [art] (ORCID: <<https://orcid.org/0000-0001-7503-8685>>),

Logo designer),

Université Bordeaux Montaigne [fnd] (ROR: <<https://ror.org/03pbgwk21>>),

CNRS [fnd] (ROR: <<https://ror.org/02feahw73>>)

Repository CRAN

Date/Publication 2026-06-22 15:50:12 UTC

Contents

benthos	3
biplot	3
bootstrap	7
burt	9
ca	10
cdt	11
colours	13
countries	13
describe	14
dimnames	15
export	15
get_contributions	16
get_coordinates	18
get_data	19
get_eigenvalues	20
mca	21
pca	22
pcoa	24
plot	26
predict	28
screeplot	29
subset	30
summary	32

benthos

3

tidy

33

viz_confidence

35

viz_contributions

38

viz_ellipses

40

viz_hull

42

viz_individuals

43

viz_tolerance

47

viz_variables

50

Index

55

benthos	<i>Benthos</i>
---------	----------------

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.carmen-n.org/?sec=data7>

See Also

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

biplot	<i>Biplot</i>
--------	---------------

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", "#E69F00"),
  col.columns = c("#56B4E9", "#56B4E9"),
  pch.rows = c(16, 1),
  pch.columns = c(17, 2),
  size = c(1, 3),
  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", "#E69F00"),
  col.columns = c("#56B4E9", "#56B4E9"),
  pch.rows = c(16, 1),
  lty.columns = c(1, 3),
  xlim = NULL,
  ylim = NULL,
  main = NULL,
  sub = NULL,
  legend = list(x = "topleft")
)
```

Arguments

<code>x</code>	A CA , MCA or PCA object.
<code>...</code>	Currently not used.
<code>axes</code>	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, col.columns	A length-two vector of color specification for the active and supplementary rows/columns.
pch.rows, pch.columns	A length-two vector of symbol specification for the active and supplementary rows/columns.
size	A length-two numeric vector giving range of possible sizes (greater than 0). Only used if type is "contribution" (CA).
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.
lty.columns	A length-two vector of line type specification for the active and supplementary columns.

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 multivariate 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 (i.e. the proximity between rows and columns cannot be directly interpreted).

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](https://doi.org/10.1111/14679876.00275).

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

See Also

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

Examples

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

## 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(iris, scale = FALSE, sup_quali = "Species")

## 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"))
```

bootstrap

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, level = c(0.68, 0.95))

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

## Plot with convex hulls
viz_columns(Y)
viz_hull(Y)

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

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

## Plot with ellipses
viz_variables(Y)
viz_tolerance(Y, 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 symmetric [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	<i>Correspondence Analysis</i>
----	--------------------------------

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,
  sup_quali = NULL,
  autodetect = FALSE
)

## 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.
sup_quali	A vector specifying the indices of the supplementary qualitative columns.
autodetect	A logical scalar: should non-numeric variables be automatically removed?

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\(\)](#), [pcoa\(\)](#), [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
get_coordinates(X, margin = 1)

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

## Get total inertia
sum(get_inertia(X))

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

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	<i>Colours</i>
---------	----------------

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	<i>Countries</i>
-----------	------------------

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](#)

`describe`*Object Description*

Description

Object Description

Usage

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

Arguments

<code>x</code>	A CA , MCA or PCA object.
<code>...</code>	Further parameters to be passed to cat() .

Value

`describe()` is called for its side-effects. Invisibly returns `x`.

Author(s)

N. Frerebeau

See Also

Other summary: [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)
```

dimnames	<i>Dimnames of an Object</i>
----------	------------------------------

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: [subset](#)

export	<i>Export Results</i>
--------	-----------------------

Description

Creates a Zip archive of all results in CSV format.

Usage

```
export(object, ...)

## S4 method for signature 'MultivariateAnalysis'
export(object, file, flags = "-r9Xjq", ...)

## S4 method for signature 'PCOA'
export(object, file, flags = "-r9Xjq", ...)
```

Arguments

object	A CA , MCA or PCA object.
...	Currently not used.
file	A character string specifying the pathname of the zip file.
flags	A character string of flags (see utils::zip()).

Author(s)

N. Frerebeau

See Also

[utils::write.csv\(\)](#), [utils::zip\(\)](#)

Other getters: [get_contributions\(\)](#), [get_coordinates\(\)](#), [get_data\(\)](#), [get_eigenvalues\(\)](#)

Examples

```
## Not run:
## Load data
data("iris")

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

## Export results
export(X, file = "results.zip")

## End(Not run)
```

get_contributions	<i>Get Contributions</i>
-------------------	--------------------------

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. 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 getters: [export\(\)](#), [get_coordinates\(\)](#), [get_data\(\)](#), [get_eigenvalues\(\)](#)

get_coordinates	<i>Get Coordinates</i>
-----------------	------------------------

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 'PCOA'
get_coordinates(x)

## 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 getters: [export\(\)](#), [get_contributions\(\)](#), [get_data\(\)](#), [get_eigenvalues\(\)](#)

Examples

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

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

## 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 AlsoOther getters: [export\(\)](#), [get_contributions\(\)](#), [get_coordinates\(\)](#), [get_eigenvalues\(\)](#)

get_eigenvalues*Get Eigenvalues*

Description

Get Eigenvalues

Usage`get_eigenvalues(x)``get_variance(x, ...)``get_distances(x, ...)``get_inertia(x, ...)`

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

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

```
## S4 method for signature 'PCOA'
get_eigenvalues(x)
```

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

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

Arguments

<code>x</code>	An object from which to get element(s) (a CA , MCA or PCA object).
<code>...</code>	Currently not used.
<code>margin</code>	A length-one numeric vector giving the subscript which the data will be returned: 1 indicates individuals/rows (the default), 2 indicates variables/columns.
<code>digits</code>	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 amount of variance explained by each (principal) component.
- `get_distance()` returns a `numeric` vector of squared distance to the centroid.
- `get_inertia()` returns a `numeric` vector giving the inertia (weighted squared distance to the centroid).

Author(s)

N. Frerebeau

See Also

Other getters: `export()`, `get_contributions()`, `get_coordinates()`, `get_data()`

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,  
  autodetect = FALSE  
)  
  
## 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.
autodetect	A logical scalar: should numeric variables be automatically removed (except sup_quanti)?

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\(\)](#), [pcoa\(\)](#), [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,
    autodetect = FALSE
)

## 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.
autodetect	A logical scalar: should non-numeric variables be automatically removed (except <code>sup_quali</code>)?

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\(\)](#), [pcoa\(\)](#), [predict\(\)](#)

Examples

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

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

## Get eigenvalues
get_eigenvalues(X)

## Get individual cos2
head(get_cos2(X, margin = 1))

## Get variable contributions
get_contributions(X, margin = 2)

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

pcoa

Principal Coordinates Analysis

Description

Computes classical (metric) multidimensional scaling.

Usage

```
pcoa(object, ...)
```

S4 method for signature 'dist'

```
pcoa(object, rank = 2)
```

Arguments

object	A distance structure .
...	Currently not used.
rank	An integer value specifying the maximal number dimension of the space which the data are to be represented in.

Value

A [PCOA](#) object.

Author(s)

N. Frerebeau

References

Gower, J. C. (1966). Some Distance Properties of Latent Root and Vector Methods Used in Multivariate Analysis. *Biometrika*, 53(3-4): 325-338. doi:[10.1093/biomet/53.34.325](https://doi.org/10.1093/biomet/53.34.325).

See Also

[stats::cmdscale\(\)](#)

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

Examples

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

## Compute euclidean distances
d <- dist(iris[, 1:4], method = "euclidean")

## Compute principal coordinates analysis
X <- pcoa(d)

## Screeplot
screeplot(X)

## Plot results
plot(X, extra_quali = iris$Species)

## Add convex hulls
plot(
  x = X,
  extra_quali = iris$Species,
  hull = TRUE
)

## Add tolerance ellipses
plot(
  x = X,
  extra_quali = iris$Species,
  ellipse = list(type = "tolerance", level = 0.95)
)
```

plot

*Plot Coordinates***Description**

Plot Coordinates

Usage

```
## S4 method for signature 'PCOA,missing'
plot(
  x,
  ...,
  axes = c(1, 2),
  labels = FALSE,
  extra_quali = NULL,
  extra_quant = NULL,
  ellipse = NULL,
  hull = FALSE,
  color = NULL,
  fill = FALSE,
  symbol = FALSE,
  size = c(1, 6),
  xlim = NULL,
  ylim = NULL,
  main = NULL,
  sub = NULL,
  ann = graphics::par("ann"),
  frame.plot = TRUE,
  panel.first = NULL,
  panel.last = NULL,
  legend = list(x = "topleft")
)
```

Arguments

x	An R object.
...	Further graphical parameters .
axes	A length-two numeric vector giving the dimensions to be plotted.
labels	A logical scalar: should labels be drawn? Labeling a large number of points can be computationally expensive and make the graph difficult to read. A selection of points to label can be provided using a list of two named elements, filter (a string specifying how to filter the labels to be drawn) and n (an integer specifying the number of labels to be drawn). See examples below.
extra_quali	An optional vector of qualitative data for aesthetics mapping.

extra_quanti	An optional vector of quantitative data for aesthetics mapping. If a single character string is passed, it must be one of "observation", "mass", "sum", "contribution" or "cos2" (see augment()).
ellipse	A list of additional arguments to be passed to viz_ellipses() ; names of the list are used as argument names. If NULL, no ellipse are displayed.
hull	A logical scalar: should convex hulls be displayed?
color	The colors for lines and points (will be mapped to extra_quanti or extra_quali; if both are set, the latter has priority). Ignored if set to FALSE. If NULL, the default color scheme will be used.
fill	The background colors for points (will be mapped to extra_quanti or extra_quali; if both are set, the latter has priority). Ignored if set to FALSE.
symbol	A vector of plotting characters or symbols (will be mapped to extra_quali). This can either be a single character or an integer code for one of a set of graphics symbols. If symbol is a named vector, then the symbols will be associated with their name within extra_quali. Ignored if set to FALSE.
size	A length-two numeric vector giving range of possible sizes (greater than 0; will be mapped to extra_quanti). Ignored if set to FALSE.
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.
ann	A logical scalar: should the default annotation (title and x and y axis labels) appear on the plot?
frame.plot	A logical scalar: should a box be drawn around 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.

Author(s)

N. Frerebeau

See AlsoOther plot methods: [biplot\(\)](#), [screeplot\(\)](#), [viz_contributions\(\)](#), [viz_individuals\(\)](#), [viz_variables\(\)](#)

predict	<i>Predict New Coordinates</i>
---------	--------------------------------

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\(\)](#), [pcoa\(\)](#)

Examples

```
## Create a matrix
A <- matrix(data = sample(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(10, 120, TRUE), nrow = 20, ncol = 6)
```

```

predict(X, Y, margin = 1)

## Predict new column coordinates
Z <- matrix(data = sample(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
)

## S4 method for signature 'PCOA'
screeplot(
  x,
  ...,
  labels = FALSE,
  limit = NULL,
  col = "grey90",
  border = "grey10"
)

```

Arguments

<code>x</code>	A CA , MCA or PCA object.
<code>...</code>	Extra parameters to be passed to graphics::barplot() .
<code>eigenvalues</code>	A logical scalar: should the eigenvalues be plotted instead of variance/inertia?
<code>cumulative</code>	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\(\)](#), [plot\(\)](#), [viz_contributions\(\)](#), [viz_individuals\(\)](#), [viz_variables\(\)](#)

Examples

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

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

## 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](#)

Examples

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

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

## 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 'MultivariateSummary'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)

## S4 method for signature 'CA'
summary(
  object,
  ...,
  axes = c(1, 2),
  margin = 1,
  active = TRUE,
  sup = TRUE,
  rank = NULL
)

## S4 method for signature 'PCA'
summary(
  object,
  ...,
  axes = c(1, 2),
  margin = 1,
  active = TRUE,
  sup = TRUE,
  rank = NULL
)
```

Arguments

x A [MultivariateSummary](#) object.

row.names	A character vector giving the row names for the data frame, or NULL.
optional	A logical scalar: should the names of the variables in the data frame be checked? If FALSE then the names of the variables in the data frame are checked to ensure that they are syntactically valid variable names and are not duplicated.
...	Currently not used.
object	A CA , MCA or PCA object.
axes	A length-two numeric vector giving the dimensions to be summarized.
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. Deprecated, use axes instead.

Author(s)

N. Frerebeau

See AlsoOther summary: [describe](#), [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: [describe](#), [summary\(\)](#)

Examples

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

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

## 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_confidence

Confidence Ellipses

Description

Plots confidence ellipses.

Usage

```
viz_confidence(x, y, ...)

wrap_confidence(x, y, ...)

## S4 method for signature 'numeric,numeric'
viz_confidence(
  x,
  y,
  ...,
  group = NULL,
  level = 0.95,
  color = NULL,
  fill = FALSE,
  symbol = FALSE
)
```

```
## S4 method for signature 'MultivariateAnalysis,missing'
viz_confidence(
  x,
  ...,
  group = NULL,
  level = 0.95,
  color = NULL,
  fill = FALSE,
  symbol = FALSE
)

## S4 method for signature 'MultivariateBootstrap,missing'
viz_confidence(
  x,
  ...,
  level = 0.95,
  color = FALSE,
  fill = FALSE,
  symbol = FALSE
)

## S4 method for signature 'PCOA,missing'
viz_confidence(
  x,
  ...,
  axes = c(1, 2),
  group = NULL,
  level = 0.95,
  color = NULL,
  fill = FALSE,
  symbol = FALSE
)

## S4 method for signature 'numeric,numeric'
wrap_confidence(x, y, group = NULL, level = 0.95)

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

## S4 method for signature 'PCOA,missing'
```

```
wrap_confidence(x, axes = c(1, 2), group = NULL, level = 0.95)
```

Arguments

x, y	A numeric vector. If y is missing, x must be an object from which to wrap observations (a CA , MCA or PCA object).
...	Further graphical parameters to be passed to <code>graphics::polygon()</code> .
group	A vector specifying the group an observation belongs to.
level	A numeric vector specifying the confidence/tolerance level.
color	The colors for borders (will be mapped to group). Ignored if set to FALSE. If NULL, the default color scheme will be used.
fill	The background colors (will be mapped to group). Ignored if set to FALSE.
symbol	A vector of symbols (will be mapped to group). Ignored if set to FALSE.
axes	A length-two numeric vector giving the dimensions for which to compute results.
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 used? If FALSE, standard coordinates are returned.

Value

`wrap_confidence()` returns a **data.frame** of envelope x and y coordinates. An extra column named `group` is added specifying the group an observation belongs to.

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

Author(s)

N. Frerebeau

See Also

Other envelopes: `viz_ellipses()`, `viz_hull()`, `viz_tolerance()`

Examples

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

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

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

## Plot confidence ellipses
```

```
col <- c("#004488", "#DDAA33", "#BB5566")
viz_rows(X, extra_quali = iris$Species, color = col)
viz_confidence(X, group = iris$Species, color = col, level = 0.95)
```

viz_contributions	<i>Visualize Contributions and cos²</i>
-------------------	--

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 = 1,
  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 <code>graphics::barplot()</code> .
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()`, `plot()`, `screeplot()`, `viz_individuals()`, `viz_variables()`

Examples

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

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

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

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

```
## Plot cos2
viz_cos2(X, axes = 1)
```

viz_ellipses

Ellipses

Description

Plots ellipses.

Usage

```
viz_ellipses(x, y, ...)
```

```
## S4 method for signature 'numeric,numeric'
```

```
viz_ellipses(
  x,
  y,
  ...,
  group = NULL,
  type = c("tolerance", "confidence"),
  level = 0.95,
  color = NULL,
  fill = FALSE,
  symbol = FALSE
)
```

```
## S4 method for signature 'MultivariateAnalysis,missing'
```

```
viz_ellipses(
  x,
  ...,
  group = NULL,
  type = c("tolerance", "confidence"),
  level = 0.95,
  color = NULL,
  fill = FALSE,
  symbol = FALSE
)
```

```
## S4 method for signature 'PCOA,missing'
```

```
viz_ellipses(
  x,
  ...,
  group = NULL,
  type = c("tolerance", "confidence"),
  level = 0.95,
```

```

    color = NULL,
    fill = FALSE,
    symbol = FALSE
  )

```

Arguments

x, y	A numeric vector. If y is missing, x must be an object from which to wrap observations (a CA , MCA or PCA object).
...	Further graphical parameters to be passed to graphics::polygon() .
group	A vector specifying the group an observation belongs to.
type	A character string specifying the ellipse to draw. It must be one of "tolerance" or "confidence". Any unambiguous substring can be given.
level	A numeric vector specifying the confidence/tolerance level.
color	The colors for borders (will be mapped to group). Ignored if set to FALSE. If NULL, the default color scheme will be used.
fill	The background colors (will be mapped to group). Ignored if set to FALSE.
symbol	A vector of symbols (will be mapped to group). Ignored if set to FALSE.

Value

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

Author(s)

N. Frerebeau

See Also

Other envelopes: [viz_confidence\(\)](#), [viz_hull\(\)](#), [viz_tolerance\(\)](#)

Examples

```

## Load data
data("iris")

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

## Plot with tolerance ellipses
col <- c("#004488", "#DDAA33", "#BB5566")
viz_rows(X, extra_quali = iris$Species, color = col)
viz_ellipses(
  x = X,
  type = "tolerance",
  level = c(0.68, 0.95),
  group = iris$Species,
  color = col
)

```

viz_hull

*Convex Hulls***Description**

Plots convex hull of a set of observations.

Usage

```

viz_hull(x, y, ...)

wrap_hull(x, y, ...)

## S4 method for signature 'numeric,numeric'
viz_hull(x, y, ..., group = NULL, color = NULL, fill = FALSE, symbol = FALSE)

## S4 method for signature 'MultivariateAnalysis,missing'
viz_hull(x, ..., group = NULL, color = NULL, fill = FALSE, symbol = FALSE)

## S4 method for signature 'MultivariateBootstrap,missing'
viz_hull(x, ..., color = FALSE, fill = FALSE, symbol = FALSE)

## S4 method for signature 'PCOA,missing'
viz_hull(x, ..., group = NULL, color = FALSE, fill = FALSE, symbol = FALSE)

## S4 method for signature 'numeric,numeric'
wrap_hull(x, y, group = NULL)

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

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

```

Arguments

x, y	A numeric vector. If y is missing, x must be an object from which to wrap observations (a CA , MCA or PCA object).
...	Further graphical parameters to be passed to graphics::polygon() .
group	A vector specifying the group an observation belongs to.
color	The colors for borders (will be mapped to group). Ignored if set to FALSE. If NULL, the default color scheme will be used.
fill	The background colors (will be mapped to group). Ignored if set to FALSE.
symbol	A vector of symbols (will be mapped to group). Ignored if set to FALSE.
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 used? If FALSE, standard coordinates are returned.

Value

`wrap_hull()` returns a [data.frame](#) of envelope x and y coordinates. An extra column named `group` is added specifying the group an observation belongs to.

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

Author(s)

N. Frerebeau

See Also

Other envelopes: [viz_confidence\(\)](#), [viz_ellipses\(\)](#), [viz_tolerance\(\)](#)

Examples

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

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

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

## Plot convex hulls
col <- c("#004488", "#DDAA33", "#BB5566")
viz_rows(X, extra_quali = iris$Species, color = col)
viz_hull(X, group = iris$Species, color = col)
```

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,  
  extra_quali = NULL,  
  extra_quanti = NULL,  
  ellipse = NULL,  
  hull = NULL,  
  color = NULL,  
  fill = FALSE,  
  symbol = FALSE,  
  size = c(1, 6),  
  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),  
  color = FALSE,  
  fill = FALSE,  
  symbol = FALSE,  
  legend = NULL  
)  
  
## S4 method for signature 'PCA'  
viz_individuals(  
  x,  
  ...,  
  axes = c(1, 2),  
  active = TRUE,  
  sup = TRUE,
```

```

labels = FALSE,
extra_quali = NULL,
extra_quant = NULL,
ellipse = NULL,
hull = NULL,
color = NULL,
fill = FALSE,
symbol = FALSE,
size = c(1, 6),
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 .
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? Labeling a large number of points can be computationally expensive and make the graph difficult to read. A selection of points to label can be provided using a list of two named elements, filter (a string specifying how to filter the labels to be drawn) and n (an integer specifying the number of labels to be drawn). See examples below.
extra_quali	An optional vector of qualitative data for aesthetics mapping.
extra_quant	An optional vector of quantitative data for aesthetics mapping. If a single character string is passed, it must be one of "observation", "mass", "sum", "contribution" or "cos2" (see augment()).
ellipse	A list of additional arguments to be passed to viz_ellipses() ; names of the list are used as argument names. If NULL, no ellipse are displayed.
hull	A logical scalar: should convex hulls be displayed?
color	The colors for lines and points (will be mapped to extra_quant or extra_quali; if both are set, the latter has priority). Ignored if set to FALSE. If NULL, the default color scheme will be used.
fill	The background colors for points (will be mapped to extra_quant or extra_quali; if both are set, the latter has priority). Ignored if set to FALSE.
symbol	A vector of plotting characters or symbols (will be mapped to extra_quali). This can either be a single character or an integer code for one of a set of graphics symbols. If symbol is a named vector, then the symbols will be associated with their name within extra_quali. Ignored if set to FALSE.

size	A length-two numeric vector giving range of possible sizes (greater than 0; will be mapped to extra_quanti). Ignored if set to FALSE.
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 <code>graphics::legend()</code> ; names of the list are used as argument names. If NULL, no legend is displayed.

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()`, `plot()`, `screeplot()`, `viz_contributions()`, `viz_variables()`

Examples

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

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

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

## Labels of the 10 individuals with highest cos2
viz_individuals(X, labels = list(filter = "cos2", n = 10))

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

## Graphical parameters
## Continuous values
viz_individuals(X, extra_quanti = iris$Petal.Length, symbol = 16, size = c(1, 2))
viz_individuals(X, extra_quanti = iris$Petal.Length, symbol = 16, size = c(1, 2),
  color = grDevices::hcl.colors(12, "RdPu"))
```

```

viz_variables(X, extra_quanti = "contribution",
             color = grDevices::hcl.colors(12, "BluGrn", rev = TRUE),
             size = c(0, 1))

## Discrete values
viz_individuals(X, extra_quali = iris$Species, symbol = 21:23)
viz_individuals(X, extra_quali = iris$Species, symbol = 21:23,
              fill = c("#004488", "#DDAA33", "#BB5566"),
              color = "black")

viz_variables(X, extra_quali = c("Petal", "Petal", "Sepal", "Sepal"),
             color = c("#EE7733", "#0077BB"),
             symbol = c(1, 3))

```

viz_tolerance	<i>Tolerance Ellipses</i>
---------------	---------------------------

Description

Plots tolerance ellipses.

Usage

```

viz_tolerance(x, y, ...)

wrap_tolerance(x, y, ...)

## S4 method for signature 'numeric,numeric'
viz_tolerance(
  x,
  y,
  ...,
  group = NULL,
  level = 0.95,
  color = NULL,
  fill = FALSE,
  symbol = FALSE
)

## S4 method for signature 'MultivariateAnalysis,missing'
viz_tolerance(
  x,
  ...,
  group = NULL,
  level = 0.95,
  color = NULL,
  fill = FALSE,

```

```

    symbol = FALSE
  )

## S4 method for signature 'MultivariateBootstrap,missing'
viz_tolerance(
  x,
  ...,
  level = 0.95,
  color = FALSE,
  fill = FALSE,
  symbol = FALSE
)

## S4 method for signature 'PCOA,missing'
viz_tolerance(
  x,
  ...,
  group = NULL,
  level = 0.95,
  color = NULL,
  fill = FALSE,
  symbol = FALSE
)

## S4 method for signature 'numeric,numeric'
wrap_tolerance(x, y, group = NULL, level = 0.95)

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

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

```

Arguments

<code>x, y</code>	A numeric vector. If <code>y</code> is missing, <code>x</code> must be an object from which to wrap observations (a CA , MCA or PCA object).
<code>...</code>	Further graphical parameters to be passed to graphics::polygon() .
<code>group</code>	A vector specifying the group an observation belongs to.
<code>level</code>	A numeric vector specifying the confidence/tolerance level.

color	The colors for borders (will be mapped to group). Ignored if set to FALSE. If NULL, the default color scheme will be used.
fill	The background colors (will be mapped to group). Ignored if set to FALSE.
symbol	A vector of symbols (will be mapped to group). Ignored if set to FALSE.
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 used? If FALSE, standard coordinates are returned.

Value

wrap_tolerance() returns a **data.frame** of envelope x and y coordinates. An extra column named group is added specifying the group an observation belongs to.

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

Author(s)

N. Frerebeau

See Also

Other envelopes: [viz_confidence\(\)](#), [viz_ellipses\(\)](#), [viz_hull\(\)](#)

Examples

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

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

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

## Plot tolerance ellipses
col <- c("#004488", "#DDAA33", "#BB5566")
viz_rows(X, extra_quali = iris$Species, color = col)
viz_tolerance(X, group = iris$Species, color = col, level = 0.95)
```

viz_variables	<i>Visualize Variables Factor Map</i>
---------------	---------------------------------------

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,  
  extra_quali = NULL,  
  extra_quant = NULL,  
  color = NULL,  
  fill = FALSE,  
  symbol = FALSE,  
  size = c(1, 6),  
  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),  
  color = FALSE,  
  fill = FALSE,  
  symbol = FALSE,  
  legend = NULL  
)
```

```
## S4 method for signature 'PCA'
viz_variables(
  x,
  ...,
  axes = c(1, 2),
  active = TRUE,
  sup = TRUE,
  labels = list(filter = "contribution", n = 10),
  extra_quali = NULL,
  extra_quanti = NULL,
  color = NULL,
  symbol = NULL,
  size = 1,
  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,
  extra_quali = NULL,
  extra_quanti = NULL,
  color = NULL,
  fill = FALSE,
  symbol = FALSE,
  size = c(1, 6),
  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),
color = FALSE,
fill = FALSE,
symbol = FALSE,
legend = NULL
)

```

Arguments

x	A CA , MCA or PCA object.
...	Further graphical parameters .
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? Labeling a large number of points can be computationally expensive and make the graph difficult to read. A selection of points to label can be provided using a list of two named elements, filter (a string specifying how to filter the labels to be drawn) and n (an integer specifying the number of labels to be drawn). See examples below.
extra_quali	An optional vector of qualitative data for aesthetics mapping.
extra_quant	An optional vector of quantitative data for aesthetics mapping. If a single character string is passed, it must be one of "observation", "mass", "sum", "contribution" or "cos2" (see augment()).
color	The colors for lines and points (will be mapped to extra_quant or extra_quali; if both are set, the latter has priority). Ignored if set to FALSE. If NULL, the default color scheme will be used.
fill	The background colors for points (will be mapped to extra_quant or extra_quali; if both are set, the latter has priority). Ignored if set to FALSE.
symbol	A vector of plotting characters or symbols (will be mapped to extra_quali). This can either be a single character or an integer code for one of a set of graphics symbols. If symbol is a named vector, then the symbols will be associated with their name within extra_quali. Ignored if set to FALSE.
size	A length-two numeric vector giving range of possible sizes (greater than 0; will be mapped to extra_quant). Ignored if set to FALSE.
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.

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\(\)](#), [plot\(\)](#), [screeplot\(\)](#), [viz_contributions\(\)](#), [viz_individuals\(\)](#)

Examples

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

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

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

## Labels of the 10 individuals with highest cos2
viz_individuals(X, labels = list(filter = "cos2", n = 10))

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

## Graphical parameters
## Continuous values
viz_individuals(X, extra_quanti = iris$Petal.Length, symbol = 16, size = c(1, 2))
viz_individuals(X, extra_quanti = iris$Petal.Length, symbol = 16, size = c(1, 2),
  color = grDevices::hcl.colors(12, "RdPu"))

viz_variables(X, extra_quanti = "contribution",
  color = grDevices::hcl.colors(12, "BluGrn", rev = TRUE),
  size = c(0, 1))

## Discrete values
viz_individuals(X, extra_quali = iris$Species, symbol = 21:23)
viz_individuals(X, extra_quali = iris$Species, symbol = 21:23,
  fill = c("#004488", "#DDAA33", "#BB5566"),
  color = "black")

viz_variables(X, extra_quali = c("Petal", "Petal", "Sepal", "Sepal"),
  color = c("#EE7733", "#0077BB"),
```

```
symbol = c(1, 3))
```

Index

- * **datasets**
 - benthos, 3
 - colours, 13
 - countries, 13
- * **envelopes**
 - viz_confidence, 35
 - viz_ellipses, 40
 - viz_hull, 42
 - viz_tolerance, 47
- * **getters**
 - export, 15
 - get_contributions, 16
 - get_coordinates, 18
 - get_data, 19
 - get_eigenvalues, 20
- * **multivariate analysis**
 - ca, 10
 - mca, 21
 - pca, 22
 - pcoa, 24
 - predict, 28
- * **mutators**
 - dimnames, 15
 - subset, 30
- * **plot methods**
 - biplot, 3
 - plot, 26
 - screeplot, 29
 - viz_contributions, 38
 - viz_individuals, 43
 - viz_variables, 50
- * **resampling methods**
 - bootstrap, 7
- * **summary**
 - describe, 14
 - summary, 32
 - tidy, 33
- * **tools**
 - burt, 9
 - cdt, 11
 - [[,CA,ANY,missing-method (subset), 30
 - [[,PCA,ANY,missing-method (subset), 30
 - array, 18
 - as.data.frame,MultivariateSummary-method (summary), 32
 - augment (tidy), 33
 - augment(), 27, 45, 52
 - augment,MultivariateAnalysis-method (tidy), 33
 - augment-method (tidy), 33
 - benthos, 3, 13
 - biplot, 3
 - biplot(), 27, 30, 39, 46, 53
 - biplot,CA-method (biplot), 3
 - biplot,PCA-method (biplot), 3
 - bootstrap, 7
 - bootstrap,CA-method (bootstrap), 7
 - bootstrap,PCA-method (bootstrap), 7
 - BootstrapCA, 7
 - BootstrapPCA, 7
 - burt, 9
 - burt(), 12
 - burt,data.frame-method (burt), 9
 - burt-method (burt), 9
 - CA, 4, 7, 10, 14–20, 28, 29, 33, 34, 37, 39, 41, 42, 45, 48, 52
 - ca, 10
 - ca(), 22, 24, 25, 28
 - ca,data.frame-method (ca), 10
 - ca,matrix-method (ca), 10
 - ca-method (ca), 10
 - cat(), 14
 - cdt, 11
 - cdt(), 9, 22
 - cdt,data.frame-method (cdt), 11
 - cdt,matrix-method (cdt), 11

- cdt-method (cdt), 11
- character, 5, 15–18, 27, 30, 31, 33, 39, 41, 45, 46, 52
- colnames, MultivariateAnalysis-method (dimnames), 15
- colours, 3, 13, 13
- countries, 3, 13, 13
- data.frame, 3, 9, 10, 12, 13, 17–19, 21–23, 28, 34, 37, 43, 49
- describe, 14, 33, 35
- describe, CA-method (describe), 14
- describe, PCA-method (describe), 14
- dim, MultivariateAnalysis-method (dimnames), 15
- dimnames, 15, 31
- dimnames, MultivariateAnalysis-method (dimnames), 15
- distance structure, 24
- export, 15
- export(), 17, 19–21
- export, MultivariateAnalysis-method (export), 15
- export, PCA-method (export), 15
- export-method (export), 15
- factor(), 9, 12
- finite, 5, 27, 46, 52
- get_contributions, 16
- get_contributions(), 16, 19–21
- get_contributions, MultivariateAnalysis-method (get_contributions), 16
- get_contributions-method (get_contributions), 16
- get_coordinates, 18
- get_coordinates(), 16, 17, 20, 21
- get_coordinates, MultivariateAnalysis-method (get_coordinates), 18
- get_coordinates, PCA-method (get_coordinates), 18
- get_coordinates-method (get_coordinates), 18
- get_correlations (get_contributions), 16
- get_correlations, PCA-method (get_contributions), 16
- get_correlations-method (get_contributions), 16
- get_cos2 (get_contributions), 16
- get_cos2, MultivariateAnalysis-method (get_contributions), 16
- get_cos2-method (get_contributions), 16
- get_data, 19
- get_data(), 16, 17, 19, 21
- get_data, MultivariateAnalysis-method (get_data), 19
- get_data-method (get_data), 19
- get_distances (get_eigenvalues), 20
- get_distances, MultivariateAnalysis-method (get_eigenvalues), 20
- get_distances-method (get_eigenvalues), 20
- get_eigenvalues, 20
- get_eigenvalues(), 16, 17, 19, 20
- get_eigenvalues, MultivariateAnalysis-method (get_eigenvalues), 20
- get_eigenvalues, PCA-method (get_eigenvalues), 20
- get_eigenvalues-method (get_eigenvalues), 20
- get_inertia (get_eigenvalues), 20
- get_inertia, MultivariateAnalysis-method (get_eigenvalues), 20
- get_inertia-method (get_eigenvalues), 20
- get_replications (get_coordinates), 18
- get_replications, BootstrapPCA-method (get_coordinates), 18
- get_replications, MultivariateBootstrap-method (get_coordinates), 18
- get_replications-method (get_coordinates), 18
- get_variance (get_eigenvalues), 20
- get_variance, MultivariateAnalysis-method (get_eigenvalues), 20
- get_variance-method (get_eigenvalues), 20
- graphical parameters, 26, 37, 41, 42, 45, 48, 52
- graphics::barplot(), 29, 39
- graphics::legend(), 5, 27, 46, 53
- graphics::polygon(), 37, 41, 42, 48
- integer, 7, 10, 20, 22–24, 30, 33, 39
- list, 5, 27, 31, 45, 46, 53
- logical, 5, 9, 10, 12, 15, 18, 22, 23, 26, 27, 29–31, 33, 34, 37, 39, 43, 45, 49, 52

- matrix, [9](#), [10](#), [22](#), [23](#), [28](#)
- MCA, [4](#), [14](#), [16–20](#), [22](#), [29](#), [33](#), [34](#), [37](#), [39](#), [41](#), [42](#), [45](#), [48](#), [52](#)
- mca, [21](#)
- mca(), [11](#), [24](#), [25](#), [28](#)
- mca, data.frame-method (mca), [21](#)
- mca, matrix-method (mca), [21](#)
- mca-method (mca), [21](#)
- MultivariateSummary, [32](#)
- numeric, [4](#), [5](#), [17](#), [18](#), [20](#), [21](#), [23](#), [26–28](#), [31](#), [33](#), [34](#), [37](#), [39](#), [41–43](#), [45](#), [46](#), [48](#), [49](#), [52](#)
- PCA, [4](#), [7](#), [14–20](#), [23](#), [28](#), [29](#), [33](#), [34](#), [37](#), [39](#), [41](#), [42](#), [45](#), [48](#), [52](#)
- pca, [22](#)
- pca(), [11](#), [22](#), [25](#), [28](#)
- pca, data.frame-method (pca), [22](#)
- pca, matrix-method (pca), [22](#)
- pca-method (pca), [22](#)
- PCOA, [25](#)
- pcoa, [24](#)
- pcoa(), [11](#), [22](#), [24](#), [28](#)
- pcoa, dist-method (pcoa), [24](#)
- pcoa-method (pcoa), [24](#)
- plot, [26](#)
- plot(), [6](#), [30](#), [39](#), [46](#), [53](#)
- plot, PCOA, missing-method (plot), [26](#)
- predict, [28](#)
- predict(), [11](#), [22](#), [24](#), [25](#)
- predict, CA-method (predict), [28](#)
- predict, MCA-method (predict), [28](#)
- predict, PCA-method (predict), [28](#)
- rownames, MultivariateAnalysis-method (dimnames), [15](#)
- screplot, [29](#)
- screplot(), [6](#), [27](#), [39](#), [46](#), [53](#)
- screplot, MultivariateAnalysis-method (screplot), [29](#)
- screplot, PCOA-method (screplot), [29](#)
- screplot-method (screplot), [29](#)
- stats::cmdscale(), [25](#)
- subset, [15](#), [30](#)
- summary, [32](#)
- summary(), [14](#), [35](#)
- summary, CA-method (summary), [32](#)
- summary, PCA-method (summary), [32](#)
- svd(), [11](#), [22](#), [24](#)
- tidy, [33](#)
- tidy(), [14](#), [33](#)
- tidy, MultivariateAnalysis-method (tidy), [33](#)
- tidy-method (tidy), [33](#)
- utils::write.csv(), [16](#)
- utils::zip(), [16](#)
- viz_columns (viz_variables), [50](#)
- viz_columns, MultivariateAnalysis-method (viz_variables), [50](#)
- viz_columns, MultivariateBootstrap-method (viz_variables), [50](#)
- viz_columns-method (viz_variables), [50](#)
- viz_confidence, [35](#)
- viz_confidence(), [41](#), [43](#), [49](#)
- viz_confidence, MultivariateAnalysis, missing-method (viz_confidence), [35](#)
- viz_confidence, MultivariateBootstrap, missing-method (viz_confidence), [35](#)
- viz_confidence, numeric, numeric-method (viz_confidence), [35](#)
- viz_confidence, PCOA, missing-method (viz_confidence), [35](#)
- viz_confidence-method (viz_confidence), [35](#)
- viz_contributions, [38](#)
- viz_contributions(), [6](#), [27](#), [30](#), [46](#), [53](#)
- viz_contributions, MultivariateAnalysis-method (viz_contributions), [38](#)
- viz_contributions-method (viz_contributions), [38](#)
- viz_cos2 (viz_contributions), [38](#)
- viz_cos2, MultivariateAnalysis-method (viz_contributions), [38](#)
- viz_cos2-method (viz_contributions), [38](#)
- viz_ellipses, [40](#)
- viz_ellipses(), [27](#), [37](#), [43](#), [45](#), [49](#)
- viz_ellipses, MultivariateAnalysis, missing-method (viz_ellipses), [40](#)
- viz_ellipses, numeric, numeric-method (viz_ellipses), [40](#)
- viz_ellipses, PCOA, missing-method (viz_ellipses), [40](#)
- viz_ellipses-method (viz_ellipses), [40](#)

viz_hull, 42
 viz_hull(), 37, 41, 49
 viz_hull, MultivariateAnalysis, missing-method (viz_hull), 42
 viz_hull, MultivariateBootstrap, missing-method (viz_hull), 42
 viz_hull, numeric, numeric-method (viz_hull), 42
 viz_hull, PCOA, missing-method (viz_hull), 42
 viz_hull-method (viz_hull), 42
 viz_individuals, 43
 viz_individuals(), 6, 27, 30, 39, 53
 viz_individuals, PCA-method (viz_individuals), 43
 viz_individuals-method (viz_individuals), 43
 viz_rows (viz_individuals), 43
 viz_rows, BootstrapCA-method (viz_individuals), 43
 viz_rows, MultivariateAnalysis-method (viz_individuals), 43
 viz_rows-method (viz_individuals), 43
 viz_tolerance, 47
 viz_tolerance(), 37, 41, 43
 viz_tolerance, MultivariateAnalysis, missing-method (viz_tolerance), 47
 viz_tolerance, MultivariateBootstrap, missing-method (viz_tolerance), 47
 viz_tolerance, numeric, numeric-method (viz_tolerance), 47
 viz_tolerance, PCOA, missing-method (viz_tolerance), 47
 viz_tolerance-method (viz_tolerance), 47
 viz_variables, 50
 viz_variables(), 6, 27, 30, 39, 46
 viz_variables, BootstrapPCA-method (viz_variables), 50
 viz_variables, CA-method (viz_variables), 50
 viz_variables, PCA-method (viz_variables), 50
 viz_variables-method (viz_variables), 50

 wrap_confidence (viz_confidence), 35
 wrap_confidence, MultivariateAnalysis, missing-method (viz_confidence), 35
 wrap_confidence, numeric, numeric-method (viz_confidence), 35
 wrap_confidence, PCOA, missing-method (viz_confidence), 35
 wrap_confidence-method (viz_confidence), 35
 wrap_hull (viz_hull), 42
 wrap_hull, MultivariateAnalysis, missing-method (viz_hull), 42
 wrap_hull, numeric, numeric-method (viz_hull), 42
 wrap_hull, PCOA, missing-method (viz_hull), 42
 wrap_hull-method (viz_hull), 42
 wrap_tolerance (viz_tolerance), 47
 wrap_tolerance, MultivariateAnalysis, missing-method (viz_tolerance), 47
 wrap_tolerance, numeric, numeric-method (viz_tolerance), 47
 wrap_tolerance, PCOA, missing-method (viz_tolerance), 47
 wrap_tolerance-method (viz_tolerance), 47