

# Package ‘dimensio’

August 23, 2023

**Title** Multivariate Data Analysis

**Version** 0.4.0

**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 (>= 3.5)

**Imports** ggplot2, graphics, grDevices, methods, rlang

**Suggests** khroma, knitr, rmarkdown, rsvg, svglite, tinysnapshot,  
tinytest

**VignetteBuilder** knitr

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.3

**Collate** 'AllClasses.R' 'AllGenerics.R' 'augment.R' 'biplot.R'  
'bootstrap.R' 'ca.R' 'coerce.R' 'data.R'  
'dimensio-deprecated.R' 'dimensio-internal.R'  
'dimensio-package.R' 'get\_contributions.R' 'get\_coordinates.R'  
'get\_correlations.R' 'get\_cos2.R' 'get\_data.R'  
'get\_distances.R' 'get\_eigenvalues.R' 'get\_inertia.R'  
'get\_variance.R' 'loadings.R' 'mutators.R' 'pca.R'  
'predicates.R' 'predict.R' 'screplot.R' 'show.R' 'subset.R'  
'summary.R' 'svd.R' 'tidy.R' 'viz\_contributions.R'

'viz\_coordinates.R' 'viz\_cos2.R' 'viz\_ellipse.R' 'viz\_hull.R'  
 'viz\_labels.R' 'wrap\_ellipses.R' 'wrap\_hull.R' 'zzz.R'

**NeedsCompilation** no

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

**Repository** CRAN

**Date/Publication** 2023-08-23 15:10:06 UTC

## R topics documented:

|                   |    |
|-------------------|----|
| benthos           | 3  |
| biplot            | 3  |
| bootstrap         | 6  |
| ca                | 8  |
| colours           | 10 |
| countries         | 10 |
| dimnames          | 11 |
| get_contributions | 11 |
| get_coordinates   | 13 |
| get_data          | 14 |
| get_distances     | 15 |
| get_eigenvalues   | 16 |
| loadings          | 17 |
| pca               | 18 |
| predict           | 20 |
| screeplot         | 21 |
| subset            | 22 |
| summary           | 24 |
| tidy              | 25 |
| viz_contributions | 26 |
| viz_individuals   | 28 |
| viz_variables     | 31 |
| viz_wrap          | 34 |
| wrap              | 35 |

**Index** 38

---

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,
  labels = "columns",
  col.rows = "#004488",
  col.columns = "#BB5566",
  cex.rows = graphics::par("cex"),
  cex.columns = graphics::par("cex"),
  pch.rows = 16,
  pch.columns = 17,
```

```

    main = NULL,
    sub = NULL,
    ...
)

## S4 method for signature 'PCA'
biplot(
  x,
  axes = c(1, 2),
  type = c("form", "covariance"),
  active = TRUE,
  sup = TRUE,
  labels = "variables",
  col.rows = "#004488",
  col.columns = "#BB5566",
  pch.rows = 16,
  pch.columns = 17,
  lty = "solid",
  lwd = 2,
  main = NULL,
  sub = NULL,
  ...
)

```

### Arguments

|                                    |   |
|------------------------------------|---|
| <code>x</code>                     | A <a href="#">CA</a> or <a href="#">PCA</a> object.   |
| <code>axes</code>                  | A length-two <a href="#">numeric</a> vector giving the dimensions to be plotted.  |
| <code>type</code>                  | A <a href="#">character</a> 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. |
| <code>active</code>                | A <a href="#">logical</a> scalar: should the active observations be plotted?  |
| <code>sup</code>                   | A <a href="#">logical</a> scalar: should the supplementary observations be plotted?   |
| <code>labels</code>                | A <a href="#">character</a> vector specifying whether "rows"/"individuals" and/or "columns"/"variables" names must be drawn. Any unambiguous substring can be given.  |
| <code>col.rows, col.columns</code> | A color specification.  |
| <code>cex.rows, cex.columns</code> | A numerical vector giving the amount by which plotting characters and symbols should be scaled relative to the default.   |
| <code>pch.rows, pch.columns</code> | A symbol specification.   |
| <code>main</code>                  | A <a href="#">character</a> string giving a main title for the plot.  |
| <code>sub</code>                   | A <a href="#">character</a> string giving a subtitle for the plot.  |
| <code>...</code>                   | Further parameters to be passed to <code>wordcloud::wordlayout()</code> .   |
| <code>lty, lwd</code>              | A specification for the line type and width.  |

## Details

A biplot is the simultaneous representation of rows and columns of a rectangular dataset. It is the generalization of a scatterplot to the case of 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` 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: [screeplot\(\)](#), [viz\\_contributions\(\)](#), [viz\\_individuals\(\)](#), [viz\\_variables\(\)](#), [viz\\_wrap](#), [wrap](#)

**Examples**

```

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

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

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

## Column-metric-preserving biplot (covariance biplot)
biplot(Y, type = "covariance")

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

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

## Row principal CA biplot
biplot(X, type = "row", labels = "columns")

## Column principal CA biplot
biplot(X, type = "column", labels = "columns")

## Contribution CA biplot
biplot(X, type = "contrib", labels = NULL)

```

---

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
viz_rows(Y)
viz_tolerance(Y, margin = 1, level = c(0.68, 0.95))

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

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

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

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

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

---

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 <a href="#">matrix</a> or a <a href="#">data.frame</a> .  |
| ...     | Currently not used.  |
| rank    | An <a href="#">integer</a> 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 <a href="#">numeric</a> or <a href="#">logical</a> vector specifying the indices of the supplementary rows.  |
| sup_col | A <a href="#">numeric</a> or <a href="#">logical</a> vector specifying the indices of the supplementary columns.   |

### Value

A [CA](#) object.

### Author(s)

N. Frerebeau



## References

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

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

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

## See Also

[svd\(\)](#)

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

## Examples

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

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

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

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

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

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

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

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

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

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

## Get eigenvalues
get_eigenvalues(X)
```

---

colours

*Colours*

---

**Description**

Contingency table of eye and hair colours of different individuals.

**Usage**

colours

**Format**

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

**Source**

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

**See Also**

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

---

countries

*Countries*

---

**Description**

Student ratings of 13 countries on six attributes.

**Usage**

countries

**Format**

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

**Source**

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

**See Also**

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

---

|          |                              |
|----------|------------------------------|
| dimnames | <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 <a href="#">CA</a> or <a href="#">PCA</a> object). |
| do.NULL | A <a href="#">logical</a> scalar. If FALSE and names are NULL, names are created.                              |
| prefix  | A <a href="#">character</a> string specifying the prefix for created names.                                    |

**Author(s)**

N. Frerebeau

**See Also**

Other mutators: [get\\_contributions\(\)](#), [get\\_coordinates\(\)](#), [get\\_data\(\)](#), [get\\_distances\(\)](#), [get\\_eigenvalues\(\)](#), [loadings\(\)](#), [subset\(\)](#)

---

|                   |                          |
|-------------------|--------------------------|
| get_contributions | <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 <a href="#">CA</a> or <a href="#">PCA</a> object).   |
| ...      | Currently not used.  |
| margin   | A length-one <a href="#">numeric</a> vector giving the subscript which the data will be returned: 1 indicates individuals/rows (the default), 2 indicates variables/columns. |
| sup_name | A <a href="#">character</a> string specifying the name of the column to create for supplementary points attribution (see below).   |

**Value**

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

**Author(s)**

N. Frerebeau

**See Also**

Other mutators: [dimnames\(\)](#), [get\\_coordinates\(\)](#), [get\\_data\(\)](#), [get\\_distances\(\)](#), [get\\_eigenvalues\(\)](#), [loadings\(\)](#), [subset\(\)](#)

---

|                 |                        |
|-----------------|------------------------|
| get_coordinates | <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 '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 <a href="#">CA</a> or <a href="#">PCA</a> object).   |
| ...       | Currently not used.  |
| margin    | A length-one <a href="#">numeric</a> vector giving the subscript which the data will be returned: 1 indicates individuals/rows (the default), 2 indicates variables/columns. |
| principal | A <a href="#">logical</a> scalar: should principal coordinates be returned? If FALSE, standard coordinates are returned.   |
| sup_name  | A <a href="#">character</a> string specifying the name of the column to create for supplementary points attribution (see below).   |

## Value

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

## Author(s)

N. Frerebeau

## See Also

Other mutators: [dimnames\(\)](#), [get\\_contributions\(\)](#), [get\\_data\(\)](#), [get\\_distances\(\)](#), [get\\_eigenvalues\(\)](#), [loadings\(\)](#), [subset\(\)](#)

## Examples

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

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

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

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

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

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

---

get\_data

*Get Original Data*

---

## Description

Get Original Data

## Usage

```
get_data(x, ...)
```

## S4 method for signature 'MultivariateAnalysis'  
get\_data(x)

## Arguments

x                    An object from which to get element(s) (a [CA](#) or [PCA](#) object).  
...                   Currently not used.

## Value

Returns a [data.frame](#) of original data.

## Author(s)

N. Frerebeau

**See Also**

Other mutators: [dimnames\(\)](#), [get\\_contributions\(\)](#), [get\\_coordinates\(\)](#), [get\\_distances\(\)](#), [get\\_eigenvalues\(\)](#), [loadings\(\)](#), [subset\(\)](#)

---

`get_distances`*Get Distances*

---

**Description**

Get Distances

**Usage**

```
get_distances(x, ...)
```

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

**Arguments**

|                     |  |
|---------------------|--|
| <code>x</code>      | An object from which to get element(s) (a <a href="#">CA</a> or <a href="#">PCA</a> object).   |
| <code>...</code>    | Currently not used.  |
| <code>margin</code> | A length-one <a href="#">numeric</a> 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 centroid.

**Author(s)**

N. Frerebeau

**See Also**

Other mutators: [dimnames\(\)](#), [get\\_contributions\(\)](#), [get\\_coordinates\(\)](#), [get\\_data\(\)](#), [get\\_eigenvalues\(\)](#), [loadings\(\)](#), [subset\(\)](#)

---

get\_eigenvalues      *Get Eigenvalues*

---

### Description

Get Eigenvalues

### Usage

```
get_eigenvalues(x)
```

```
get_inertia(x, ...)
```

```
get_variance(x, ...)
```

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

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

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

### Arguments

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

### Value

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

### Author(s)

N. Frerebeau



**See Also**

Other mutators: [dimnames\(\)](#), [get\\_contributions\(\)](#), [get\\_coordinates\(\)](#), [get\\_data\(\)](#), [get\\_distances\(\)](#), [loadings\(\)](#), [subset\(\)](#)

---

loadings

*Extract Loadings*

---

**Description**

Extract loadings in principal components analysis.

**Usage**

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

**Arguments**

x                    A [PCA](#) object.

**Value**

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

**Note**

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

**Author(s)**

N. Frerebeau

**See Also**

Other mutators: [dimnames\(\)](#), [get\\_contributions\(\)](#), [get\\_coordinates\(\)](#), [get\\_data\(\)](#), [get\\_distances\(\)](#), [get\\_eigenvalues\(\)](#), [subset\(\)](#)

**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 <a href="#">matrix</a> or a <a href="#">data.frame</a> .   |
| ...     | Currently not used.   |
| center  | A <a href="#">logical</a> scalar: should the variables be shifted to be zero centered?  |
| scale   | A <a href="#">logical</a> scalar: should the variables be scaled to unit variance?  |
| rank    | An <a href="#">integer</a> 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 <a href="#">numeric</a> or <a href="#">logical</a> vector specifying the indices of the supplementary rows (individuals).   |

|            |  |
|------------|--|
| sup_col    | A <a href="#">numeric</a> or <a href="#">logical</a> vector specifying the indices of the supplementary columns (variables).   |
| weight_row | A <a href="#">numeric</a> 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 <a href="#">numeric</a> 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 correlations between variables and dimensions
head(get_correlations(X))

## Get eigenvalues
get_eigenvalues(X)
```

predict

*Predict New Coordinates*

---

**Description**

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

**Usage**

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

**Arguments**

|         |   |
|---------|---|
| object  | A <a href="#">CA</a> or <a href="#">PCA</a> object.   |
| newdata | An object of supplementary points coercible to a <a href="#">matrix</a> for which to compute principal coordinates.   |
| margin  | A length-one <a href="#">numeric</a> 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)
```

---

screepLOT

*Scree Plot*


---

## Description

Plot eigenvalues (scree plot) or variances histogram.

## Usage

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

## Arguments

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

## Value

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

**Author(s)**

N. Frerebeau

**See Also**

Other plot methods: [biplot\(\)](#), [viz\\_contributions\(\)](#), [viz\\_individuals\(\)](#), [viz\\_variables\(\)](#), [viz\\_wrap](#), [wrap](#)

**Examples**

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

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

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

---

subset

*Extract Parts of an Object*

---

**Description**

Operators acting on objects to extract parts.

**Usage**

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

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

**Arguments**

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

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

**Details**

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

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

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

sd A **numeric** vector giving the variables standard deviations (PCA).

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

coord A **numeric** matrix of rows/individuals coordinates.

cos2 A **numeric** matrix of rows/individuals squared cosine.

masses A **numeric** vector giving the rows masses/individual weights.

sup A **logical** vector specifying whether a point is a supplementary observation or not.

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

coord A **numeric** matrix of columns/variables coordinates.

cor A **numeric** matrix of correlation between variables and the dimensions (PCA).

cos2 A **numeric** matrix of columns/variables squared cosine.

masses A **numeric** vector giving the columns masses/variable weights.

sup A **logical** vector specifying whether a point is a supplementary observation or not.

If *i* is "eigenvalues", returns a **numeric** vector of eigenvalues.

## Value

A **list**.

## Author(s)

N. Frerebeau

## See Also

Other mutators: [dimnames\(\)](#), [get\\_contributions\(\)](#), [get\\_coordinates\(\)](#), [get\\_data\(\)](#), [get\\_distances\(\)](#), [get\\_eigenvalues\(\)](#), [loadings\(\)](#)

## Examples

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

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

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

---

summary

*Object Summaries*

---

### Description

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

### Usage

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

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

### Arguments

|        |  |
|--------|--|
| object | A <a href="#">CA</a> or <a href="#">PCA</a> object.  |
| margin | A length-one <a href="#">numeric</a> vector giving the subscript which the data will be summarized: 1 indicates individuals/rows (the default), 2 indicates variables/columns. |
| active | A <a href="#">logical</a> scalar: should the active observations be summarized?  |
| sup    | A <a href="#">logical</a> scalar: should the supplementary observations be summarized?   |
| rank   | An <a href="#">integer</a> 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 | <i>Tidy Coordinates</i> |
|------|-------------------------|

---

**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 <a href="#">CA</a> or <a href="#">PCA</a> object.  |
| ...       | Currently not used.  |
| margin    | A length-one <a href="#">numeric</a> vector giving the subscript which the data will be returned: 1 indicates individuals/rows (the default), 2 indicates variables/columns. |
| axes      | A length-two <a href="#">numeric</a> vector giving the dimensions to be for which to compute results.  |
| principal | A <a href="#">logical</a> 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

### Examples

```

## Load data
data("iris")

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

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

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

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

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

```

---

viz\_contributions      *Visualize Contributions and  $\cos^2$*

---

### Description

Plots contributions histogram and  $\cos^2$  scatterplot.

### Usage

```

viz_contributions(x, ...)

viz_cos2(x, ...)

## S4 method for signature 'MultivariateAnalysis'
viz_contributions(
  x,
  margin = 2,

```

```

    axes = 1,
    sort = TRUE,
    decreasing = TRUE,
    limit = 10,
    horiz = FALSE,
    col = "grey90",
    border = "grey10",
    ...
)

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

```

### Arguments

|             |  |
|-------------|--|
| x           | A <a href="#">CA</a> or <a href="#">PCA</a> object.  |
| ...         | Extra parameters to be passed to <a href="#">graphics::barplot()</a> .   |
| margin      | A length-one <a href="#">numeric</a> vector giving the subscript which the data will be returned: 1 indicates individuals/rows (the default), 2 indicates variables/columns. |
| axes        | A length-one <a href="#">numeric</a> vector giving the dimensions to be plotted.   |
| sort        | A <a href="#">logical</a> scalar: should the data be sorted?   |
| decreasing  | A <a href="#">logical</a> scalar: should the sort order be decreasing? Only used if sort is TRUE.  |
| limit       | An <a href="#">integer</a> specifying the number of top elements to be displayed.  |
| horiz       | A <a href="#">logical</a> scalar: should the bars be drawn horizontally with the first at the bottom?  |
| col, border | A <a href="#">character</a> string specifying the bars infilling and border colors.  |
| active      | A <a href="#">logical</a> scalar: should the active observations be plotted?   |
| sup         | A <a href="#">logical</a> scalar: should the supplementary observations be plotted?  |

### Value

viz\_contributions() and viz\_cos2() are called for their side-effects: they result in a graphic being displayed. Invisibly return x.

**Author(s)**

N. Frerebeau

**See Also**

Other plot methods: [biplot\(\)](#), [screeplot\(\)](#), [viz\\_individuals\(\)](#), [viz\\_variables\(\)](#), [viz\\_wrap](#), [wrap](#)

**Examples**

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

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

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

## Plot contributions
viz_contributions(X)

## Plot cos2
viz_cos2(X)
```

---

viz\_individuals

*Visualize Individuals Factor Map*

---

**Description**

Plots row/individual principal coordinates.

**Usage**

```
viz_individuals(x, ...)

viz_rows(x, ...)

## S4 method for signature 'MultivariateAnalysis'
viz_rows(
  x,
  axes = c(1, 2),
  active = TRUE,
  sup = TRUE,
  labels = FALSE,
  highlight = NULL,
  main = NULL,
  sub = NULL,
```

```

    panel.first = NULL,
    panel.last = NULL,
    ...
)

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

## S4 method for signature 'PCA'
viz_individuals(
  x,
  axes = c(1, 2),
  active = TRUE,
  sup = TRUE,
  labels = FALSE,
  highlight = NULL,
  main = NULL,
  sub = NULL,
  panel.first = NULL,
  panel.last = NULL,
  ...
)

```

### Arguments

|             |  |
|-------------|--|
| x           | A <a href="#">CA</a> or <a href="#">PCA</a> object.  |
| ...         | Further <a href="#">graphical parameters</a> (see details).  |
| axes        | A length-two <a href="#">numeric</a> vector giving the dimensions to be plotted.   |
| active      | A <a href="#">logical</a> scalar: should the active observations be plotted?   |
| sup         | A <a href="#">logical</a> scalar: should the supplementary observations be plotted?  |
| labels      | A <a href="#">logical</a> scalar: should labels be drawn?  |
| highlight   | A vector specifying the information to be highlighted. If NULL (the default), no highlighting is applied. It will only be mapped if at least one <a href="#">graphical parameters</a> is explicitly specified (see examples). If a single character string is passed, it must be one of "observation", "mass", "sum", "contribution" or "cos2" (see <a href="#">augment()</a> ). Any unambiguous substring can be given. |
| main        | A <a href="#">character</a> string giving a main title for the plot.   |
| sub         | A <a href="#">character</a> string giving a subtitle for the plot.   |
| panel.first | An 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.   |

### Details

Commonly used [graphical parameters](#) are:

- pch** A vector of plotting characters or symbols. This can either be a single character or an integer code for one of a set of graphics symbols.
- cex** A numerical vector giving the amount by which plotting characters and symbols should be scaled relative to the default.
- col** The colors for lines and points. Multiple colors can be specified so that each point can be given its own color.
- bg** The background color for the open plot symbols given by `pch = 21:25`.

**Value**

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

**Note**

Be careful: graphical parameters are silently recycled.

**Author(s)**

N. Frerebeau

**See Also**

Other plot methods: [biplot\(\)](#), [screeplot\(\)](#), [viz\\_contributions\(\)](#), [viz\\_variables\(\)](#), [viz\\_wrap](#), [wrap](#)

**Examples**

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

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

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

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

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

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

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

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

---

|               |                                       |
|---------------|---------------------------------------|
| viz_variables | <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,
  highlight = NULL,
  main = NULL,
  sub = NULL,
  panel.first = NULL,
  panel.last = NULL,
  ...
)

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

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

```

    active = TRUE,
    sup = TRUE,
    labels = TRUE,
    highlight = NULL,
    main = NULL,
    sub = NULL,
    panel.first = NULL,
    panel.last = NULL,
    ...
)

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

```

### Arguments

|             |  |
|-------------|--|
| x           | A <a href="#">CA</a> or <a href="#">PCA</a> object.  |
| ...         | Further <a href="#">graphical parameters</a> (see details).  |
| axes        | A length-two <a href="#">numeric</a> vector giving the dimensions to be plotted.   |
| active      | A <a href="#">logical</a> scalar: should the active observations be plotted?   |
| sup         | A <a href="#">logical</a> scalar: should the supplementary observations be plotted?  |
| labels      | A <a href="#">logical</a> scalar: should labels be drawn?  |
| highlight   | A vector specifying the information to be highlighted. If <code>NULL</code> (the default), no highlighting is applied. It will only be mapped if at least one <a href="#">graphical parameter</a> is explicitly specified (see examples). If a single character string is passed, it must be one of "observation", "mass", "sum", "contribution" or "cos2" (see <a href="#">augment()</a> ). Any unambiguous substring can be given. |
| main        | A <a href="#">character</a> string giving a main title for the plot.   |
| sub         | A <a href="#">character</a> 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.   |

### Details

Commonly used [graphical parameters](#) are:

|     |  |
|-----|--|
| pch | A vector of plotting characters or symbols. This can either be a single character or an integer code for one of a set of graphics symbols. |
| cex | A numerical vector giving the amount by which plotting characters and symbols should be scaled relative to the default.                    |
| lty | A vector of line types.  |
| lwd | A vector of line widths.   |
| col | The colors for lines and points. Multiple colors can be specified so that each point can be given its own color.                           |
| bg  | The background color for the open plot symbols given by <code>pch = 21:25</code> .   |



**Value**

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

**Note**

Be careful: graphical parameters are silently recycled.

**Author(s)**

N. Frerebeau

**See Also**

Other plot methods: [biplot\(\)](#), [screplot\(\)](#), [viz\\_contributions\(\)](#), [viz\\_individuals\(\)](#), [viz\\_wrap](#), [wrap](#)

**Examples**

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

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

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

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

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

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

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

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

viz\_wrap

*Plot Envelopes***Description**

Plot Envelopes

**Usage**

viz\_hull(x, ...)

viz\_confidence(x, ...)

viz\_tolerance(x, ...)

## S4 method for signature 'MultivariateAnalysis'

viz\_tolerance(x, margin = 1, axes = c(1, 2), group = NULL, level = 0.95, ...)

## S4 method for signature 'BootstrapCA'

viz\_tolerance(x, margin = 1, axes = c(1, 2), level = 0.95, ...)

## S4 method for signature 'MultivariateAnalysis'

viz\_confidence(x, margin = 1, axes = c(1, 2), group = NULL, level = 0.95, ...)

## S4 method for signature 'BootstrapCA'

viz\_confidence(x, margin = 1, axes = c(1, 2), level = 0.95, ...)

## S4 method for signature 'MultivariateAnalysis'

viz\_hull(x, margin = 1, axes = c(1, 2), group = NULL, ...)

## S4 method for signature 'BootstrapCA'

viz\_hull(x, margin = 1, axes = c(1, 2), ...)

**Arguments**

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

**Value**

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

**Author(s)**

N. Frerebeau

**See Also**

Other plot methods: [biplot\(\)](#), [screplot\(\)](#), [viz\\_contributions\(\)](#), [viz\\_individuals\(\)](#), [viz\\_variables\(\)](#), [wrap](#)

**Examples**

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

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

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

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

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

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

---

wrap

*Wrap Observations*

---

**Description**

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

**Usage**

```

wrap_hull(x, ...)

wrap_confidence(x, ...)

wrap_tolerance(x, ...)

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

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

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

```

**Arguments**

|        |  |
|--------|--|
| x      | An object from which to wrap observations (a <a href="#">CA</a> or <a href="#">PCA</a> object).  |
| ...    | Currently not used.  |
| margin | A length-one <a href="#">numeric</a> vector giving the subscript which the data will be returned: 1 indicates individuals/rows (the default), 2 indicates variables/columns. |
| axes   | A length-two <a href="#">numeric</a> vector giving the dimensions to be for which to compute results.  |
| group  | A vector specifying the group an observation belongs to.   |
| level  | A <a href="#">numeric</a> vector specifying the confidence/tolerance level.  |

**Value**

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

**Author(s)**

N. Frerebeau

**See Also**

Other plot methods: [biplot\(\)](#), [screeplot\(\)](#), [viz\\_contributions\(\)](#), [viz\\_individuals\(\)](#), [viz\\_variables\(\)](#), [viz\\_wrap](#)

**Examples**

```

## Load data
data("iris")

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

```

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

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

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

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

# Index

- \* **datasets**
  - benthos, 3
  - colours, 10
  - countries, 10
- \* **multivariate analysis**
  - ca, 8
  - pca, 18
  - predict, 20
- \* **mutators**
  - dimnames, 11
  - get\_contributions, 11
  - get\_coordinates, 13
  - get\_data, 14
  - get\_distances, 15
  - get\_eigenvalues, 16
  - loadings, 17
  - subset, 22
- \* **plot methods**
  - biplot, 3
  - screeplot, 21
  - viz\_contributions, 26
  - viz\_individuals, 28
  - viz\_variables, 31
  - viz\_wrap, 34
  - wrap, 35
- \* **resampling methods**
  - bootstrap, 6
- \* **summary**
  - summary, 24
- \* **tidy methods**
  - tidy, 25
- [[, CA, ANY, missing-method (subset), 22
- [[, PCA, ANY, missing-method (subset), 22
- array, 13
- augment (tidy), 25
- augment(), 29, 32
- augment, MultivariateAnalysis-method (tidy), 25
- augment-method (tidy), 25
- benthos, 3, 10
- biplot, 3, 22, 28, 30, 33, 35, 36
- biplot, CA-method (biplot), 3
- biplot, PCA-method (biplot), 3
- bootstrap, 6
- bootstrap, CA-method (bootstrap), 6
- bootstrap, PCA-method (bootstrap), 6
- bootstrap-method (bootstrap), 6
- BootstrapCA, 7
- BootstrapPCA, 7
- CA, 4, 7, 8, 11–16, 20, 21, 24, 25, 27, 29, 32, 34, 36
- ca, 8, 19, 20
- ca, data.frame-method (ca), 8
- ca, matrix-method (ca), 8
- ca-method (ca), 8
- character, 4, 11–13, 21, 22, 27, 29, 32
- colnames, MultivariateAnalysis-method (dimnames), 11
- colours, 3, 10, 10
- countries, 3, 10, 10
- data.frame, 3, 8, 10, 12–14, 16, 18, 20, 25, 36
- dim, MultivariateAnalysis-method (dimnames), 11
- dimnames, 11, 12, 13, 15, 17, 23
- dimnames, MultivariateAnalysis-method (dimnames), 11
- get\_contributions, 11, 11, 13, 15, 17, 23
- get\_contributions, MultivariateAnalysis-method (get\_contributions), 11
- get\_contributions-method (get\_contributions), 11
- get\_coordinates, 11, 12, 13, 15, 17, 23
- get\_coordinates, MultivariateAnalysis-method (get\_coordinates), 13
- get\_coordinates-method (get\_coordinates), 13

- get\_correlations (get\_contributions), 11
- get\_correlations, PCA-method (get\_contributions), 11
- get\_correlations-method (get\_contributions), 11
- get\_cos2 (get\_contributions), 11
- get\_cos2, MultivariateAnalysis-method (get\_contributions), 11
- get\_cos2-method (get\_contributions), 11
- get\_data, 11–13, 14, 15, 17, 23
- get\_data, MultivariateAnalysis-method (get\_data), 14
- get\_data-method (get\_data), 14
- get\_distances, 11–13, 15, 15, 17, 23
- get\_distances, MultivariateAnalysis-method (get\_distances), 15
- get\_distances-method (get\_distances), 15
- get\_eigenvalues, 11–13, 15, 16, 17, 23
- get\_eigenvalues, MultivariateAnalysis-method (get\_eigenvalues), 16
- get\_eigenvalues-method (get\_eigenvalues), 16
- get\_inertia (get\_eigenvalues), 16
- get\_inertia, MultivariateAnalysis-method (get\_eigenvalues), 16
- get\_inertia-method (get\_eigenvalues), 16
- get\_replications (get\_coordinates), 13
- get\_replications, BootstrapPCA-method (get\_coordinates), 13
- get\_replications, MultivariateBootstrap-method (get\_coordinates), 13
- get\_replications-method (get\_coordinates), 13
- get\_variance (get\_eigenvalues), 16
- get\_variance, MultivariateAnalysis-method (get\_eigenvalues), 16
- get\_variance-method (get\_eigenvalues), 16
- graphical parameters, 29, 32, 34
- graphics::barplot(), 21, 27
- graphics::polygon(), 34
- integer, 7, 8, 16, 18, 21, 24, 27
- list, 23
- loadings, 11–13, 15, 17, 17, 23
- loadings, PCA-method (loadings), 17
- logical, 4, 8, 11, 13, 18, 19, 21, 23–25, 27, 29, 32
- matrix, 8, 18, 20
- numeric, 4, 8, 12, 13, 15, 16, 18–20, 22–25, 27, 29, 32, 34, 36
- PCA, 4, 7, 11–17, 19–21, 24, 25, 27, 29, 32, 34, 36
- pca, 9, 18, 20
- pca, data.frame-method (pca), 18
- pca, matrix-method (pca), 18
- pca-method (pca), 18
- predict, 9, 19, 20
- predict, CA-method (predict), 20
- predict, PCA-method (predict), 20
- rownames, MultivariateAnalysis-method (dimnames), 11
- screepplot, 5, 21, 28, 30, 33, 35, 36
- screepplot, MultivariateAnalysis-method (screepplot), 21
- screepplot-method (screepplot), 21
- subset, 11–13, 15, 17, 22
- summary, 24
- summary, CA-method (summary), 24
- summary, PCA-method (summary), 24
- svd(), 9, 19
- tidy, 25
- tidy, MultivariateAnalysis-method (tidy), 25
- tidy-method (tidy), 25
- viz\_columns (viz\_variables), 31
- viz\_columns, BootstrapCA-method (viz\_variables), 31
- viz\_columns, MultivariateAnalysis-method (viz\_variables), 31
- viz\_columns-method (viz\_variables), 31
- viz\_confidence (viz\_wrap), 34
- viz\_confidence, BootstrapCA-method (viz\_wrap), 34
- viz\_confidence, MultivariateAnalysis-method (viz\_wrap), 34
- viz\_confidence-method (viz\_wrap), 34
- viz\_contributions, 5, 22, 26, 30, 33, 35, 36
- viz\_contributions, MultivariateAnalysis-method (viz\_contributions), 26
- viz\_contributions-method (viz\_contributions), 26

`viz_cos2` (`viz_contributions`), 26  
`viz_cos2`, `MultivariateAnalysis`-method  
    (`viz_contributions`), 26  
`viz_cos2`-method (`viz_contributions`), 26  
`viz_hull` (`viz_wrap`), 34  
`viz_hull`, `BootstrapCA`-method (`viz_wrap`),  
    34  
`viz_hull`, `MultivariateAnalysis`-method  
    (`viz_wrap`), 34  
`viz_hull`-method (`viz_wrap`), 34  
`viz_individuals`, 5, 22, 28, 28, 33, 35, 36  
`viz_individuals`, `PCA`-method  
    (`viz_individuals`), 28  
`viz_individuals`-method  
    (`viz_individuals`), 28  
`viz_rows` (`viz_individuals`), 28  
`viz_rows`, `BootstrapCA`-method  
    (`viz_individuals`), 28  
`viz_rows`, `MultivariateAnalysis`-method  
    (`viz_individuals`), 28  
`viz_rows`-method (`viz_individuals`), 28  
`viz_tolerance` (`viz_wrap`), 34  
`viz_tolerance`, `BootstrapCA`-method  
    (`viz_wrap`), 34  
`viz_tolerance`, `MultivariateAnalysis`-method  
    (`viz_wrap`), 34  
`viz_tolerance`-method (`viz_wrap`), 34  
`viz_variables`, 5, 22, 28, 30, 31, 35, 36  
`viz_variables`, `BootstrapPCA`-method  
    (`viz_variables`), 31  
`viz_variables`, `PCA`-method  
    (`viz_variables`), 31  
`viz_variables`-method (`viz_variables`), 31  
`viz_wrap`, 5, 22, 28, 30, 33, 34, 36  
  
`wordcloud::wordlayout()`, 4  
`wrap`, 5, 22, 28, 30, 33, 35, 35  
`wrap_confidence` (`wrap`), 35  
`wrap_confidence`, `MultivariateAnalysis`-method  
    (`wrap`), 35  
`wrap_confidence`-method (`wrap`), 35  
`wrap_hull` (`wrap`), 35  
`wrap_hull`, `MultivariateAnalysis`-method  
    (`wrap`), 35  
`wrap_hull`-method (`wrap`), 35  
`wrap_tolerance` (`wrap`), 35  
`wrap_tolerance`, `MultivariateAnalysis`-method  
    (`wrap`), 35  
`wrap_tolerance`-method (`wrap`), 35