

Package ‘booklet’

July 22, 2025

Type Package

Title Multivariate Exploratory Data Analysis

Version 1.0.0

Description Exploratory data analysis methods to summarize, visualize and describe datasets. The main principal component methods are available, those with the largest potential in terms of applications: principal component analysis (PCA) when variables are quantitative, correspondence analysis (CA) when variables are categorical, Multiple Factor Analysis (MFA) when variables are structured in groups.

License MIT + file LICENSE

URL <https://github.com/alexym1/booklet>,
<https://alexym1.github.io/booklet/>

BugReports <https://github.com/alexym1/booklet/issues>

Depends R (>= 4.1.0)

Suggests covr, devtools, factoextra, FactoMineR, knitr, renv, testthat

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author Alex Yahiaoui Martinez [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-5315-675X>>)

Maintainer Alex Yahiaoui Martinez <yahiaoui-martinez.alex@outlook.com>

Repository CRAN

Date/Publication 2025-04-24 07:10:02 UTC

Contents

ca_col_contrib	2
ca_col_coords	3
ca_col_cos2	4

ca_col_inertia	4
ca_row_contrib	5
ca_row_coords	6
ca_row_cos2	6
ca_row_inertia	7
ca_standardize	8
ca_weighted_eigen	8
facto_ca	9
facto_mfa	10
facto_pca	10
pca_eigen	11
pca_ind_contrib	12
pca_ind_coords	13
pca_ind_cos2	14
pca_standardize_norm	14
pca_var_contrib	15
pca_var_coords	16
pca_var_cor	17
pca_var_cos2	17
Index	19

ca_col_contrib	<i>Compute col contributions</i>
----------------	----------------------------------

Description

Return col contributions for each correspondence component

Usage

ca_col_contrib(col_coords,X, eigs)

Arguments

- col_coords col coordinates
- X standardized matrix
- eigs eigs computed by ca_weighted_eigen

Value

A dataframe of col contributions.

Examples

```
library(booklet)

X_scaled <- mtcars[, c(2, 8:11)] |>
  ca_standardize()

eigs <- X_scaled |>
  ca_weighted_eigen()

eigs |>
  ca_col_coords() |>
  ca_col_contrib(X_scaled, eigs) |>
  head()
```

ca_col_coords	<i>Compute col coordinates</i>
---------------	--------------------------------

Description

Return Correspondence component for columns

Usage

```
ca_col_coords(eigs)

ca_col_sup_coords(X_sup, eigs)
```

Arguments

eigs	eigs computed by ca_weighted_eigen
X_sup	Supplementary dataset

Value

A dataframe of col coordinates.

Examples

```
library(booklet)

mtcars[, c(2, 8:11)] |>
  ca_standardize() |>
  ca_weighted_eigen() |>
  ca_col_coords() |>
  head()
```

ca_col_cos2	<i>Compute col squared cosines</i>
-------------	------------------------------------

Description

Return col squared cosines for each correspondence component

Usage

```
ca_col_cos2(col_coords, X)

ca_col_sup_cos2(col_coords, X_sup, X)
```

Arguments

col_coords	col coordinates
X	active dataset
X_sup	supplementary dataset

Value

A dataframe of col squared cosines.

Examples

```
library(booklet)

X_scaled <- mtcars[, c(2, 8:11)] |>
  ca_standardize()

X_scaled |>
  ca_weighted_eigen() |>
  ca_col_coords() |>
  ca_col_cos2(X_scaled) |>
  head()
```

ca_col_inertia	<i>Compute col inertia</i>
----------------	----------------------------

Description

Return col inertia for each correspondence component

Usage

```
ca_col_inertia(X)
```

Arguments

X standardized matrix

Value

A dataframe of col inertia.

Examples

```
library(booklet)

mtcars[, c(2, 8:11)] |>
  ca_standardize() |>
  ca_col_inertia()
```

ca_row_contrib	<i>Compute row contributions</i>
----------------	----------------------------------

Description

Return row contributions for each correspondence component

Usage

```
ca_row_contrib(row_coords, X, eigs)
```

Arguments

row_coords row coordinates
 X standardized matrix
 eigs eigs computed by ca_weighted_eigen

Value

A dataframe of row contributions.

Examples

```
library(booklet)

X_scaled <- mtcars[, c(2, 8:11)] |>
  ca_standardize()

eigs <- X_scaled |>
  ca_weighted_eigen()

eigs |>
  ca_row_coords() |>
  ca_row_contrib(X_scaled, eigs) |>
  head()
```

ca_row_coords	<i>Compute row coordinates</i>
---------------	--------------------------------

Description

Return Correspondence component for individuals

Usage

```
ca_row_coords(eigs)

ca_row_sup_coords(X_sup, eigs)
```

Arguments

eigs	eigs computed by ca_weighted_eigen
X_sup	Supplementary dataset

Value

A dataframe of row coordinates.

Examples

```
library(booklet)

mtcars[, c(2, 8:11)] |>
  ca_standardize() |>
  ca_weighted_eigen() |>
  ca_row_coords() |>
  head()
```

ca_row_cos2	<i>Compute row squared cosines</i>
-------------	------------------------------------

Description

Return row squared cosines for each correspondence component

Usage

```
ca_row_cos2(row_coords, X)

ca_row_sup_cos2(row_coords, X_sup, X)
```

Arguments

row_coords	row coordinates
X	Active standardized matrix
X_sup	Supplementary standardized matrix

Value

A dataframe of row squared cosines.

Examples

```
library(booklet)

X_scaled <- mtcars[, c(2, 8:11)] |>
  ca_standardize()

X_scaled |>
  ca_weighted_eigen() |>
  ca_row_coords() |>
  ca_row_cos2(X_scaled) |>
  head()
```

ca_row_inertia	<i>Compute row inertia</i>
----------------	----------------------------

Description

Return row inertia for each correspondence component

Usage

```
ca_row_inertia(X)
```

Arguments

X	standardized matrix
---	---------------------

Value

A dataframe of row inertia.

Examples

```
library(booklet)

mtcars[, c(2, 8:11)] |>
  ca_standardize() |>
  ca_row_inertia()
```

ca_standardize	<i>Data standardization for CA</i>
----------------	------------------------------------

Description

Perform data standardization for multivariate exploratory data analysis.

Usage

```
ca_standardize(X, weighted_row = rep(1, nrow(X)))
```

```
ca_standardize_sup(X, type = c("row", "col"), weighted_row = rep(1, nrow(X)))
```

Arguments

X	Active or supplementary datasets
weighted_row	row weights
type	standardization for supplementary rows or cols

Value

A dataframe of the same size as X.

Examples

```
library(booklet)

mtcars[, c(2, 8:11)] |>
  ca_standardize() |>
  head()
```

ca_weighted_eigen	<i>Compute eigenvalues and eigenvectors for CA</i>
-------------------	--

Description

Return eigenvalues and eigenvectors of a matrix

Usage

```
ca_weighted_eigen(X)
```

Arguments

X	X_active
---	----------

Value

A list containing results of Single Value Decomposition (SVD).

Examples

```
library(booklet)

mtcars[, c(2, 8:11)] |>
  ca_standardize() |>
  ca_weighted_eigen() |>
  head()
```

facto_ca

Perform CA with FactoMineR's style

Description

Return CA results with FactoMineR's style

Usage

```
facto_ca(X, ncp = 5, row_sup = NULL, col_sup = NULL, weighted_row = NULL)
```

Arguments

X	a data frame with n rows (individuals) and p columns (numeric variables)
ncp	an integer, the number of components to keep (value set by default)
row_sup	a vector indicating the indexes of the supplementary rows
col_sup	a vector indicating the indexes of the supplementary cols
weighted_row	row weights

Value

A list containing results of FactoMineR's correspondence analysis (CA).

Examples

```
library(booklet)
res <- facto_ca(X = mtcars[, c(2, 8:11)], ncp = 2)
```

facto_mfa	<i>Perform MFA with FactoMineR's style</i>
-----------	--

Description

Return MFA results with FactoMineR's style

Usage

```
facto_mfa(X, groups, ncp = 2)
```

Arguments

X	a data frame with n rows (individuals) and p columns (numeric variables)
groups	a vector indicating the group of each variable
ncp	an integer, the number of components to keep (value set by default)

Value

A list containing results of FactoMineR's multiple factor analysis (MFA).

Examples

```
library(booklet)

res <- facto_mfa(X = iris[, -c(5)], groups = c(2, 2), ncp = 2)
```

facto_pca	<i>Perform PCA with FactoMineR's style</i>
-----------	--

Description

Return PCA results with FactoMineR's style

Usage

```
facto_pca(
  X,
  ncp = 5,
  scale.unit = TRUE,
  ind_sup = NULL,
  quanti_sup = NULL,
  weighted_col = NULL
)
```

Arguments

<code>X</code>	a data frame with <code>n</code> rows (individuals) and <code>p</code> columns (numeric variables)
<code>ncp</code>	an integer, the number of components to keep (value set by default)
<code>scale.unit</code>	a boolean, if TRUE (value set by default) then data are scaled to unit variance
<code>ind_sup</code>	a vector indicating the indexes of the supplementary individuals
<code>quanti_sup</code>	a vector indicating the indexes of the quantitative supplementary variables
<code>weighted_col</code>	column weights

Value

A list containing results of FactoMineR's principal components analysis (PCA).

Examples

```
library(booklet)

res <- facto_pca(iris[, -5], ncp = 2, ind_sup = 1, quanti_sup = 1)
```

<code>pca_eigen</code>	<i>Compute eigenvalues and eigenvectors</i>
------------------------	---

Description

Return eigenvalues and eigenvectors of a matrix

Usage

```
pca_eigen(X)

pca_weighted_eigen(
  X,
  weighted_row = rep(1, nrow(X))/nrow(X),
  weighted_col = rep(1, ncol(X))
)
```

Arguments

<code>X</code>	<code>X_active</code>
<code>weighted_row</code>	row weights
<code>weighted_col</code>	column weights

Details

Standardization depends on what you need to perform factor analysis. We implemented two types:

- `pca_weighted_eigen`: This is the default method in FactoMineR to compute eigvalues, eigvectors and U matrix.
- `pca_eigen`: This is the standard method to compute eigenvalues, eigenvectors.

Value

A list containing results of Single Value Decomposition (SVD).

Examples

```
library(booklet)

iris[, -5] |>
  pca_standardize_norm() |>
  pca_eigen()
```

pca_ind_contrib	<i>Compute individual contributions</i>
-----------------	---

Description

Return individual contributions for each principal component

Usage

```
pca_ind_contrib(
  ind_coords,
  eigs,
  weighted_row = rep(1, nrow(ind_coords))/nrow(ind_coords)
)
```

Arguments

ind_coords	individual coordinates
eigs	eigs computed by <code>pca_eigen</code> or <code>pca_weighted_eigen</code>
weighted_row	row weights

Details

If you want to compute the contributions of the individuals to the principal components, you have to change the `weighted_col` argument to `rep(1, nrow(ind_cos2))`.

Value

A dataframe of individual contributions.

Examples

```
library(booklet)

eigs <- iris[, -5] |>
  pca_standardize_norm() |>
  pca_weighted_eigen()

eigs |>
  pca_ind_coords() |>
  pca_ind_contrib(eigs) |>
  head()
```

pca_ind_coords	<i>Compute coordinates for individuals</i>
----------------	--

Description

Return principal component for individuals

Usage

```
pca_ind_coords(eigs)
```

Arguments

eigs eigs computed by `pca_eigen` or `pca_weighted_eigen`

Value

A dataframe of individual coordinates.

Examples

```
library(booklet)

iris[, -5] |>
  pca_standardize_norm() |>
  pca_weighted_eigen() |>
  pca_ind_coords() |>
  head()
```

pca_ind_cos2	<i>Compute individual squared cosines</i>
--------------	---

Description

Return individual squared cosines for each principal component

Usage

```
pca_ind_cos2(ind_coords, weighted_col = rep(1, ncol(ind_coords)))
```

Arguments

ind_coords	individual coordinates
weighted_col	column weights

Value

A dataframe of individual squared cosines.

Examples

```
library(booklet)

iris[, -5] |>
  pca_standardize_norm() |>
  pca_weighted_eigen() |>
  pca_ind_coords() |>
  pca_ind_cos2() |>
  head()
```

pca_standardize_norm	<i>Data standardization for PCA</i>
----------------------	-------------------------------------

Description

Perform data standardization for multivariate exploratory data analysis.

Usage

```
pca_standardize_norm(X, center = TRUE, scale = TRUE)

pca_standardize(X, scale = TRUE, weighted_row = rep(1, nrow(X))/nrow(X))
```

Arguments

X	matrix
center	centering by the mean
scale	scaling by the standard deviation
weighted_row	row weights

Details

Standardization depends on what you need to perform factor analysis. Two methods are implemented:

- `standardize`: standardization is performed by centering the data matrix and dividing by the square root of the sum of squares of the weights. This is the same method used in `FactoMineR::PCA()`.
- `standardize_norm`: standardization is performed by centering and scaling the data matrix. $(X - \mu) / S$, where μ is the mean and S is the standard deviation.

Value

A dataframe of the same size as X.

Examples

```
library(booklet)

iris[, -5] |>
  pca_standardize_norm() |>
  head()
```

pca_var_contrib	<i>Compute variable contributions</i>
-----------------	---------------------------------------

Description

Return variable contributions

Usage

```
pca_var_contrib(var_cos2, eigs, weighted_col = rep(1, ncol(var_cos2)))
```

Arguments

var_cos2	variable coordinates
eigs	eigs computed by <code>pca_eigen</code> or <code>pca_weighted_eigen</code>
weighted_col	column weights

Value

A dataframe of variable contributions.

Examples

```
library(booklet)

eigs <- iris[, -5] |>
  pca_standardize_norm() |>
  pca_weighted_eigen()

eigs |>
  pca_var_coords() |>
  pca_var_cos2() |>
  pca_var_contrib(eigs) |>
  head()
```

pca_var_coords	<i>Compute variable coordinates</i>
----------------	-------------------------------------

Description

Return variable coordinates

Usage

```
pca_var_coords(eigs)
```

Arguments

eigs eigs computed by pca_eigen or pca_weighted_eigen

Value

A dataframe of variable coordinates.

Examples

```
library(booklet)

iris[, -5] |>
  pca_standardize_norm() |>
  pca_weighted_eigen() |>
  pca_var_coords() |>
  head()
```

pca_var_cor	<i>Compute variable correlation</i>
-------------	-------------------------------------

Description

Return variable correlation

Usage

```
pca_var_cor(eigs)
```

Arguments

eigs eigs computed by pca_eigen or pca_weighted_eigen

Value

A dataframe of variable correlation.

Examples

```
library(booklet)

iris[, -5] |>
  pca_standardize_norm() |>
  pca_weighted_eigen() |>
  pca_var_cor() |>
  head()
```

pca_var_cos2	<i>Compute variable squared cosines</i>
--------------	---

Description

Return variable squared cosines

Usage

```
pca_var_cos2(var_coords)
```

Arguments

var_coords variable coordinates

Value

A dataframe of variable squared consines.

Examples

```
library(booklet)

iris[, -5] |>
  pca_standardize_norm() |>
  pca_weighted_eigen() |>
  pca_var_coords() |>
  pca_var_cos2() |>
  head()
```

Index

ca_col_contrib, [2](#)
ca_col_coords, [3](#)
ca_col_cos2, [4](#)
ca_col_inertia, [4](#)
ca_col_sup_coords (ca_col_coords), [3](#)
ca_col_sup_cos2 (ca_col_cos2), [4](#)
ca_row_contrib, [5](#)
ca_row_coords, [6](#)
ca_row_cos2, [6](#)
ca_row_inertia, [7](#)
ca_row_sup_coords (ca_row_coords), [6](#)
ca_row_sup_cos2 (ca_row_cos2), [6](#)
ca_standardize, [8](#)
ca_standardize_sup (ca_standardize), [8](#)
ca_weighted_eigen, [8](#)

facto_ca, [9](#)
facto_mfa, [10](#)
facto_pca, [10](#)

pca_eigen, [11](#)
pca_ind_contrib, [12](#)
pca_ind_coords, [13](#)
pca_ind_cos2, [14](#)
pca_standardize (pca_standardize_norm),
 [14](#)
pca_standardize_norm, [14](#)
pca_var_contrib, [15](#)
pca_var_coords, [16](#)
pca_var_cor, [17](#)
pca_var_cos2, [17](#)
pca_weighted_eigen (pca_eigen), [11](#)