

Package ‘spca’

August 31, 2026

Type Package

Title Least Squares Sparse Principal Components Analysis

Version 1.1.3

Date 2026-08-29

Description Implements least-squares sparse principal component analysis with cardinality constraints. The package has an efficient C++ backend and provides functions for fitting, summarizing, comparing, and visualizing sparse principal component models. The approach follows Merola (2015) <[doi:10.1111/anzs.12128](https://doi.org/10.1111/anzs.12128)> and Merola and Chen (2019) <[doi:10.1016/j.jmva.2019.04.001](https://doi.org/10.1016/j.jmva.2019.04.001)>.

Maintainer Giovanni Maria Merola <merolagio@gmail.com>

URL <https://github.com/merolagio/spca>

BugReports <https://github.com/merolagio/spca/issues>

License AGPL-3

Encoding UTF-8

Depends R (>= 4.3)

Imports Rcpp (>= 1.0.14), ggplot2 (>= 4.0.0), RMTstat (>= 0.3.1), scales, rlang

Suggests testthat (>= 3.0.0), peakRAM (>= 1.0.2), knitr, rmarkdown, R.rsp

VignetteBuilder knitr

Config/testthat/edition 3

LinkingTo Rcpp, RcppEigen

RoxygenNote 7.3.2

LazyData true

NeedsCompilation yes

Author Giovanni Maria Merola [aut, cre]

Repository CRAN

Date/Publication 2026-08-31 15:40:02 UTC

Contents

spca-package	2
aggregate_by_group	4
change_loadings_sign_spca	5
change_sign	5
change_weights_sign_spca	6
compare_spca	7
holzinger	9
holzinger_scales	10
is.spca	10
new_spca	11
pca	12
plot.spca	13
print.spca	16
qqplot_spca	17
screeplot_spca	19
show_correlations	20
show_weights	21
spca	22
spca_object	25
spca_screeplot	26
summary.spca	27
wachter_qqplot	28
Index	30

spca-package	<i>Least Squares Sparse Principal Components Analysis</i>
--------------	---

Description

The package provides functions to compute LS-SPCA solutions, in which sparsity is imposed on Pearson PCA’s least-squares reconstruction objective.

Details

LS-SPCA differs from SPCA methods that compute sparse PCs by maximizing variance. Details are provided in the references below and in the extended vignette.

This release accompanies the related article and supports reproduction of the results reported therein.

Computation relies on efficient C++ routines and includes multiple options for variable selection and sparse weight estimation.

Fitting functions

- `spca()` Computes LS-SPCA solutions from a data or covariance/correlation matrix. It returns a `spca_object` of class `spca`.

- `pca()` Computes PCA solutions from a data or covariance/correlation matrix. It returns a `spca_object` inheriting from classes `spca_pca` and `spca`.

Methods

- `print()`, `summary()`, and `plot()` inspect and display `spca` objects.
- `change_sign()` changes the signs of selected components and their related object elements.
- `show_weights()` prints or returns the nonzero weights or contributions for selected components.
- `aggregate_by_group()` aggregates weights or contributions according to a grouping vector.
- `screeplot_spca()` and `qqplot_spca()` provide diagnostic plots for objects returned by `pca()`.

Utilities

- `is.spca()` Verifies whether an object is an `spca` object.
- `compare_spca()` Compares two or more LS-SPCA solutions numerically and visually.
- `new_spca()` Creates an `spca` object from a set of weights.

The former interfaces `change_weights_sign_spca()`, `change_loadings_sign_spca()`, `spca_screeplot()`, and `wachter_qqplot()` are retained for backward compatibility and issue deprecation warnings. Objects created by previous package versions with `loadings` and `loadings_list` elements remain supported.

Author(s)

Maintainer: Giovanni Maria Merola <merolagio@gmail.com>

References

- Merola, G. M. (2015). Least Squares Sparse Principal Component Analysis: a Backward Elimination approach to attain large weights. *Australia & New Zealand Journal of Statistics*, 57, 391–429. doi:10.1111/anzs.12128
- Merola, G. M. and Chen, G. (2019). Projection sparse principal component analysis: An efficient least squares method. *Journal of Multivariate Analysis*, 173, 366–382. doi:10.1016/j.jmva.2019.04.001

See Also

Useful links:

- <https://github.com/merolagio/spca>
- Report bugs at <https://github.com/merolagio/spca/issues>

aggregate_by_group	<i>Aggregate SPCA Weights or Contributions by Group</i>
--------------------	---

Description

Aggregate component weights or contributions according to a grouping variable.

Usage

```
aggregate_by_group(spca_obj, ...)

## S3 method for class 'spca'
aggregate_by_group(
  spca_obj,
  groups,
  only_nonzero = TRUE,
  contributions = TRUE,
  digits = ifelse(contributions, 1, 3),
  print_table = TRUE,
  return_table = FALSE,
  ...
)
```

Arguments

<code>spca_obj</code>	An object of class <code>spca</code> .
<code>...</code>	Additional arguments reserved for S3 method compatibility.
<code>groups</code>	A vector or factor with one group label per variable.
<code>only_nonzero</code>	A logical value indicating whether to omit groups whose values are zero in every selected component.
<code>contributions</code>	A logical value. If <code>TRUE</code> , aggregate percentage contributions; otherwise, aggregate weights.
<code>digits</code>	Number of digits used in printed output.
<code>print_table</code>	A logical value indicating whether to print the table.
<code>return_table</code>	A logical value indicating whether to return the table visibly.

Value

The aggregated matrix, visibly when `return_table = TRUE` and invisibly otherwise.

See Also

Other `spca`: [change_loadings_sign_spca\(\)](#), [change_sign\(\)](#), [change_weights_sign_spca\(\)](#), [compare_spca\(\)](#), [is.spca\(\)](#), [new_spca\(\)](#), [plot.spca\(\)](#), [print.spca\(\)](#), [show_correlations\(\)](#), [show_weights\(\)](#), [spca\(\)](#), [spca_object](#), [summary.spca\(\)](#)

change_loadings_sign_spca

Change Component Signs in an SPCA Object (Deprecated Alias)

Description

change_loadings_sign_spca() is retained for backward compatibility. Use [change_sign\(\)](#) in new code.

Usage

```
change_loadings_sign_spca(spca_obj, index_to_change)
```

Arguments

spca_obj An object of class spca.

index_to_change

An integer vector of component indices whose signs should be changed.

Value

The modified spca_obj.

See Also

Other spca: [aggregate_by_group\(\)](#), [change_sign\(\)](#), [change_weights_sign_spca\(\)](#), [compare_spca\(\)](#), [is.spca\(\)](#), [new_spca\(\)](#), [plot.spca\(\)](#), [print.spca\(\)](#), [show_correlations\(\)](#), [show_weights\(\)](#), [spca\(\)](#), [spca_object](#), [summary.spca\(\)](#)

change_sign

Change Component Signs

Description

Change the signs of selected components in a fitted object.

Usage

```
change_sign(spca_obj, ...)
```

```
## S3 method for class 'spca'
```

```
change_sign(spca_obj, index_to_change, ...)
```

Arguments

spca_obj An object of class spca.
 ... Additional arguments reserved for S3 method compatibility.
 index_to_change An integer vector of component indices whose signs should be changed.

Value

The modified object.
 The modified spca object.

See Also

Other spca: [aggregate_by_group\(\)](#), [change_loadings_sign_spca\(\)](#), [change_weights_sign_spca\(\)](#), [compare_spca\(\)](#), [is.spca\(\)](#), [new_spca\(\)](#), [plot.spca\(\)](#), [print.spca\(\)](#), [show_correlations\(\)](#), [show_weights\(\)](#), [spca\(\)](#), [spca_object](#), [summary.spca\(\)](#)

change_weights_sign_spca

Change Component Signs in an SPCA Object (Deprecated Alias)

Description

`change_weights_sign_spca()` is retained for backward compatibility. Use [change_sign\(\)](#) in new code.

Usage

```
change_weights_sign_spca(spca_obj, index_to_change)
```

Arguments

spca_obj An object of class spca.
 index_to_change An integer vector of component indices whose signs should be changed.

Value

The modified spca_obj.

See Also

Other spca: [aggregate_by_group\(\)](#), [change_loadings_sign_spca\(\)](#), [change_sign\(\)](#), [compare_spca\(\)](#), [is.spca\(\)](#), [new_spca\(\)](#), [plot.spca\(\)](#), [print.spca\(\)](#), [show_correlations\(\)](#), [show_weights\(\)](#), [spca\(\)](#), [spca_object](#), [summary.spca\(\)](#)

Description

Plots weights and print summary statistics for two or more sPCA objects side by side. For the meaning of each summary statistic, see [summary.sPCA](#). Tables and plots can optionally be returned.

Usage

```
compare_sPCA(
  obj_list,
  n_comps = NULL,
  contributions = TRUE,
  only_nonzero = TRUE,
  variable_groups = NULL,
  plot_weights = TRUE,
  plot_type = c("bars", "points"),
  methods_names = NULL,
  x_axis_var_names = FALSE,
  col_grouplines = "red",
  color_scale = c("ggplot", "cbb", "printsafe", "bw"),
  col_short_names = TRUE,
  print_tables = TRUE,
  print_weights = FALSE,
  show_plot = TRUE,
  return_tables = FALSE,
  return_plot = FALSE
)
```

Arguments

<code>obj_list</code>	A list of two or more sPCA objects.
<code>n_comps</code>	An integer scalar or NULL (default NULL). Number of components to compare. If NULL, the minimum number of available components across objects is used.
<code>contributions</code>	A logical value (default TRUE). If TRUE, compare percentage contributions; otherwise, compare weights.
<code>only_nonzero</code>	A logical value (default TRUE). If TRUE, only variables with at least one nonzero weight or contribution are plotted or printed.
<code>variable_groups</code>	Optional variable grouping (default NULL). Can be a list of indices, a vector, or a factor with one entry per variable. Used to draw vertical group-separating lines in the weights plot.
<code>plot_weights</code>	A logical value (default TRUE). If TRUE, plot the weights or contributions.
<code>plot_type</code>	A character vector (default first element "bars"). Values starting with "b" use bars; values starting with "p" use points. Other values default to bars.

methods_names	An optional character vector (default NULL) with one label per object. If NULL, labels are M1, ..., Mk.
x_axis_var_names	A logical value (default FALSE). If TRUE, show variable names on the x axis of the weights plot.
col_grouplines	A character scalar (default "red"). Color of the vertical group lines.
color_scale	A character vector (default first element "ggplot"). Color palette for bar plots. Accepted values are "ggplot", "cbb", "printsafe", and "bw".
col_short_names	A logical value (default TRUE). If TRUE, use short component names such as C1.M1; otherwise, use names such as C1.object_name.
print_tables	A logical value (default TRUE). If FALSE, suppress table printing. Takes priority over print_weights.
print_weights	A logical value (default FALSE). If TRUE, print the weights or contributions table.
show_plot	A logical value (default TRUE). If TRUE, show the weights or contributions plot.
return_tables	A logical value (default FALSE). If TRUE, return the weights or contributions matrix and the raw summary matrix.
return_plot	A logical value (default FALSE). If TRUE, return the weights or contributions plot.

Value

Invisibly returns NULL by default. If return_tables = TRUE, returns a list containing the comparison matrix and summary matrix. If return_plot = TRUE, the returned object also includes the weights or contributions plot.

See Also

Other spca: [aggregate_by_group\(\)](#), [change_loadings_sign_spca\(\)](#), [change_sign\(\)](#), [change_weights_sign_spca\(\)](#), [is.spca\(\)](#), [new_spca\(\)](#), [plot.spca\(\)](#), [print.spca\(\)](#), [show_correlations\(\)](#), [show_weights\(\)](#), [spca\(\)](#), [spca_object](#), [summary.spca\(\)](#)

Examples

```
data(holzinger)
ho_uspca = spca(holzinger, n_comps = 4, method = "u")
ho_cspca = spca(holzinger, n_comps = 4, method = "c")
compare_spca(list(ho_uspca, ho_cspca))
```

holzinger*Holzinger–Swineford Student Ability Data*

Description

This dataset is based on the classic Holzinger and Swineford (1939) Student Ability dataset.

We use the version distributed with the `psychTools` package. For comparability with previous analyses, we select 12 items and only students from the Grant–White School (see also Ferrara, Martella, and Vichi, 2019).

Usage

```
holzinger
```

Format

holzinger A numeric data frame with 145 rows and 12 variables. The variables are:

visual Visual perception test (SPL).

cubes Cubes test (SPL).

flags Lozenges test (SPL).

paragraph Paragraph comprehension test (VBL).

sentence Sentence completion test (VBL).

wordm Word meaning test (VBL).

addition Addition test (SPD).

counting Counting groups of dots test (SPD).

straight Straight and curved capitals test (SPD).

deduct Deduction test (MTH).

numeric Numerical puzzles test (MTH).

series Series completion test (MTH).

Details

The 12 items correspond to four ability scales: spatial (SPL), verbal (VBL), speed (SPD), and mathematical (MTH). The data provided with this package are scaled to mean zero and unit variance. The scales are available as a factor called `holzinger_scales`.

References

- Holzinger, K. J., and Swineford, F. (1939). *A study in factor analysis: The stability of a bi-factor solution*. Supplementary Educational Monographs, No. 48.
- Ferrara, C., Martella, F., and Vichi, M. (2019). Probabilistic disjoint principal component analysis. *Multivariate Behavioral Research*, 54(1), 47–61.

holzinger_scales	<i>Holzinger–Swineford Student Ability Scales</i>
------------------	---

Description

Holzinger–Swineford Student Ability Scales

Usage

holzinger_scales

Format

holzinger_scales A factor listing the 4 scales: SPL, VBL, SPD and MTH, for each variable.

is.spca	<i>Test for SPCA Objects</i>
---------	------------------------------

Description

Check whether an object has class spca and contains the core elements required by the package.

Usage

is.spca(x)

Arguments

x An object to test.

Details

The function checks for class spca and for the presence of the core elements used by the package, including weights, contributions, explained-variance summaries, component counts, cardinalities, weight lists, and active indices. It performs a lightweight structural check; use validate_spca() for a more detailed internal validation.

Value

A logical value. Returns TRUE if x has class spca and contains the required core elements, and FALSE otherwise.

See Also

Other spca: [aggregate_by_group\(\)](#), [change_loadings_sign_spca\(\)](#), [change_sign\(\)](#), [change_weights_sign_spca\(\)](#), [compare_spca\(\)](#), [new_spca\(\)](#), [plot.spca\(\)](#), [print.spca\(\)](#), [show_correlations\(\)](#), [show_weights\(\)](#), [spca\(\)](#), [spca_object](#), [summary.spca\(\)](#)

Examples

```
data(holzinger)
ho_cspca = spca(holzinger, n_comps = 2)
is.spca(ho_cspca)
```

new_spca	<i>Construct an SPCA Object from a Set of Weights</i>
----------	---

Description

Build an object of class `spca` from a weights matrix and either a covariance or correlation matrix, a data matrix, or both.

Usage

```
new_spca(A, S = NULL, X = NULL, method_name = NULL)
```

Arguments

A	A numeric matrix of weights.
S	A numeric covariance or correlation matrix (default NULL). If NULL, X is used to estimate the covariance matrix.
X	A numeric data matrix or data frame (default NULL). Used to compute S when S = NULL, and to compute scores when supplied. At least one of S or X must be provided.
method_name	A character scalar or NULL (default NULL). Name of the method used to compute the weights.

Value

An `spca` object.

See Also

Other `spca`: [aggregate_by_group\(\)](#), [change_loadings_sign_spca\(\)](#), [change_sign\(\)](#), [change_weights_sign_spca\(\)](#), [compare_spca\(\)](#), [is.spca\(\)](#), [plot.spca\(\)](#), [print.spca\(\)](#), [show_correlations\(\)](#), [show_weights\(\)](#), [spca\(\)](#), [spca_object](#), [summary.spca\(\)](#)

Examples

```
set.seed(1)
A = round(matrix(runif(24, -1, 1), 12))
A[abs(A) < 0.4] = 0 #no need to scale to unit norm
data(holzinger)
spca_new = new_spca(A, X = holzinger)
is.spca(spca_new)
summary(spca_new)
```

pca

*Computes Principal Components***Description**

Compute a principal component analysis (PCA) and return the result as an `spca` object, so that it can be used with `spca` methods.

Usage

```
pca(
  M,
  n_comps = NULL,
  center_data = FALSE,
  scale_data = FALSE,
  fat_matrix = NULL,
  screeplot = FALSE,
  qq_plot = TRUE,
  nrow_data = NULL,
  neigen_toplot = NULL,
  cor = TRUE,
  common_var = 1,
  pm = FALSE,
  eps_pm = 1e-04,
  maxiter_pm = 1000
)
```

Arguments

<code>M</code>	A data matrix, correlation matrix, or covariance matrix.
<code>n_comps</code>	An integer scalar or <code>NULL</code> (default <code>NULL</code>). Number of components to retain. If <code>NULL</code> , all components are retained up to the maximum allowed by the selected backend.
<code>center_data</code>	A logical value (default <code>FALSE</code>). If <code>TRUE</code> , center variables to zero mean. If <code>M</code> is detected as a data matrix and any column mean is nonzero, centering is performed automatically.
<code>scale_data</code>	A logical value (default <code>FALSE</code>). If <code>TRUE</code> , scale variables.
<code>fat_matrix</code>	A logical value or <code>NULL</code> (default <code>NULL</code>). If <code>NULL</code> , the backend is selected automatically: data matrices with $n < p$ use the fat backend and all other inputs use the tall backend. If <code>TRUE</code> , request the fat backend. If <code>FALSE</code> , use the tall backend. Covariance and correlation matrices always use the tall backend.
<code>screeplot</code>	A logical value (default <code>FALSE</code>). If <code>TRUE</code> , produce a scree plot.
<code>qq_plot</code>	A logical value (default <code>TRUE</code>). If <code>TRUE</code> , produce a Wachter QQ plot with qqplot_spca .
<code>nrow_data</code>	An integer scalar or <code>NULL</code> (default <code>NULL</code>). Number of rows in the original data set. Required when <code>qq_plot = TRUE</code> and <code>M</code> is a covariance or correlation matrix. If not available, the Wachter QQ-plot cannot be produced.

neigen_toplot	An integer scalar or NULL (default NULL). Number of eigenvalues to show in diagnostic plots. If NULL, all available eigenvalues are shown.
cor	A logical value (default TRUE). Currently accepted for compatibility; the diagnostic plot uses common_var for the Marchenko–Pastur quantiles.
common_var	A numeric scalar (default 1). Common variance of the variables used for the Marchenko–Pastur quantiles in the Wachter QQ plot.
pm	A logical value (default FALSE). If TRUE, compute the requested eigenpairs by power method and rank-one deflation.
eps_pm	A positive numeric scalar (default 1e-4). Convergence tolerance for the power method.
maxiter_pm	A positive integer scalar (default 1000). Maximum number of power-method iterations.

Details

n_comps controls how many components are retained in the returned object. The tall backend computes PCA from the covariance or correlation matrix. The fat backend computes PCA in row space and converts the retained eigenvectors back to variable weights.

Value

An [spca_object](#) with an additional eigenvalues vector containing the eigenvalues up to the rank used by the selected backend and n_obs stores the number of observations, if a data matrix is passed, or NULL.

See Also

Other pca: [qqplot_spca\(\)](#), [screeplot_spca\(\)](#), [spca_screeplot\(\)](#), [wachter_qqplot\(\)](#)

Examples

```
data(holzinger)
ho_pca = pca(holzinger, n_comps = 4, screeplot = TRUE,
             nrow_data = 144, qq_plot = TRUE)
summary(ho_pca)
```

plot.spca

Plot an spca Object

Description

Plot the sparse weights, or the corresponding percentage contributions, from an spca object. The plot can be shown as a bar plot, circular bar plot, or heatmap.

Usage

```
## S3 method for class 'spca'
plot(
  x,
  n_plot = NULL,
  plot_type = c("bars", "circular", "heatmap"),
  contributions = TRUE,
  only_nonzero = TRUE,
  pc_weights = NULL,
  variable_groups = NULL,
  plot_title = NULL,
  return_plot = FALSE,
  show_plot = TRUE,
  controls = list(color_scale = c("ggplot", "cbb", "printsafe", "bw"), variable_names =
    "none", legend_position = c("none", "bottom", "right", "top", "left"), grid_type =
    c("horizontal", "full", "none"), facet_labels = NULL, legend_title = NULL, x_axis_lab
    = "variables", adjust_labels_circ = NULL, flip_heatmap = FALSE, heatmap_color_range =
    c("values", "unit")),
  ...
)
```

Arguments

<code>x</code>	An object of class <code>spca</code> .
<code>n_plot</code>	An integer scalar or <code>NULL</code> (default <code>NULL</code>). Number of components to plot. If <code>NULL</code> , all components in <code>x</code> are plotted.
<code>plot_type</code>	A character vector (default first element <code>"bars"</code>). Plot type. Accepted values are <code>"bars"</code> , <code>"circular"</code> , and <code>"heatmap"</code> . The first character is enough for matching.
<code>contributions</code>	A logical value (default <code>TRUE</code>). If <code>TRUE</code> , plot percentage contributions; otherwise, plot L2 unit weights.
<code>only_nonzero</code>	A logical value (default <code>TRUE</code>). If <code>TRUE</code> , plot only variables with at least one nonzero weight.
<code>pc_weights</code>	A numeric matrix, data frame, or <code>NULL</code> (default <code>NULL</code>). Optional PCA weights or contributions to plot together with the SPCA values for comparison.
<code>variable_groups</code>	A vector, factor, or <code>NULL</code> (default <code>NULL</code>). Optional grouping variable of length p , where p is the number of variables. If supplied, bars or tiles are colored by group instead of by component.
<code>plot_title</code>	A character scalar or <code>NULL</code> (default <code>NULL</code>). Optional plot title.
<code>return_plot</code>	A logical value (default <code>FALSE</code>). If <code>TRUE</code> , return the <code>ggplot2</code> object.
<code>show_plot</code>	A logical value (default <code>TRUE</code>). If <code>TRUE</code> , print the plot.
<code>controls</code>	A list of graphical controls (default described below). Supported entries are <code>color_scale</code> , <code>variable_names</code> , <code>legend_position</code> , <code>grid_type</code> , <code>facet_labels</code> , <code>legend_title</code> , <code>x_axis_lab</code> , <code>adjust_labels_circ</code> , <code>flip_heatmap</code> , and <code>heatmap_color_range</code> .
<code>...</code>	Further arguments. These are currently unused and trigger an error if supplied.

Details

If `pc_weights` is supplied, SPCA and PCA values are plotted side by side for comparison. Circular bar plots are not implemented for this comparison, so a standard bar plot is used instead. In this case all variables are plotted, regardless of `only_nonzero`.

For character arguments defined by a default vector of accepted values, the first element is the default and the first character of the supplied string is used for matching.

The entries in `controls` are:

- `color_scale`: a character vector (default first element "ggplot"). Accepted values are "ggplot", "cbb", "printsafe", and "bw". "cbb" is colorblind-friendly with black, "printsafe" is colorblind- and printer-friendly, "bw" uses gray tones, and "ggplot" uses the default ggplot2 scale.
- `variable_names`: a character vector or NULL (default "none"). If NULL, row names of the weight matrix are used, or V1, ..., Vp if row names are missing. If set to "none", variable names are not shown. If a character vector of length *p* is supplied, it is used as the variable names.
- `legend_position`: a character vector (default first element "none"). Accepted values are "none", "bottom", "right", "top", and "left".
- `grid_type`: a character vector (default first element "horizontal"). Accepted values are "horizontal", "full", and "none".
- `facet_labels`: a character vector or NULL (default NULL). Optional facet labels for components.
- `legend_title`: a character scalar or NULL (default NULL). Optional legend title.
- `x_axis_lab`: a character scalar (default "variables"). Label for the x axis.
- `adjust_labels_circ`: a numeric vector or NULL (default NULL). Optional angular adjustments for circular plot labels.
- `flip_heatmap`: a logical value (default FALSE). If TRUE, flip the heatmap axes.
- `heatmap_color_range`: a character vector (default first element "values"). Accepted values are "values" and "unit".

When variable groups are supplied, a legend is needed to identify the groups; if the legend is missing or suppressed, it is moved to the bottom. For circular plots, the legend is moved to the right unless it is suppressed.

Value

If `return_plot = TRUE`, returns the ggplot2 object. Otherwise, returns NULL invisibly.

References

The printsafe palette corresponds to OrRd from <https://colorbrewer2.org/>.

See Also

Other `spca`: [aggregate_by_group\(\)](#), [change_loadings_sign_spca\(\)](#), [change_sign\(\)](#), [change_weights_sign_spca\(\)](#), [compare_spca\(\)](#), [is.spca\(\)](#), [new_spca\(\)](#), [print.spca\(\)](#), [show_correlations\(\)](#), [show_weights\(\)](#), [spca\(\)](#), [spca_object](#), [summary.spca\(\)](#)

Examples

```
data(holzinger)
ho_cspca = spca(holzinger, n_comps = 4)
ho_plot = plot(ho_cspca, return_plot = TRUE)

# Change faceting and legend position.
ho_plot + ggplot2::facet_wrap(
  facets = ggplot2::vars(component),
  ncol = 4,
  nrow = 1
) + ggplot2::theme(legend.position = "right")
```

print.spca	<i>Print an spca Object</i>
------------	-----------------------------

Description

Print sparse weights, or the corresponding percentage contributions, from an `spca` object. By default, variables with only zero entries are omitted, and cumulative explained variance is shown at the bottom of the table.

Usage

```
## S3 method for class 'spca'
print(
  x,
  cols = NULL,
  only_nonzero = TRUE,
  contributions = TRUE,
  digits = 3,
  thresh_card = 1e-07,
  return_table = FALSE,
  component_names = NULL,
  ...
)
```

Arguments

<code>x</code>	An object of class <code>spca</code> .
<code>cols</code>	An integer vector or <code>NULL</code> (default <code>NULL</code>). Components to print. If <code>NULL</code> , all components are printed. If a single integer is supplied, components <code>1:cols</code> are printed.
<code>only_nonzero</code>	A logical value (default <code>TRUE</code>). If <code>TRUE</code> , print only variables with at least one weight or contribution whose absolute value is greater than or equal to <code>thresh_card</code> .
<code>contributions</code>	A logical value (default <code>TRUE</code>). If <code>TRUE</code> , print weights scaled to unit L_1 norm as percentage contributions; otherwise, print L_2 unit weights.

<code>digits</code>	An integer scalar (default 3). Number of decimal places used when printing weights. Contributions are printed as percentages with one decimal place.
<code>thresh_card</code>	A numeric scalar (default $1e-07$). Values with absolute magnitude below this threshold are treated as zero in the printed table.
<code>return_table</code>	A logical value (default FALSE). If TRUE, return the formatted character matrix.
<code>component_names</code>	A character vector or NULL (default NULL). Optional component names. If NULL, existing column names are used when available; otherwise default names are assigned.
<code>...</code>	Further arguments. These are currently unused and trigger an error if supplied.

Value

If `return_table = TRUE`, returns the formatted character matrix. Otherwise, returns NULL invisibly.

See Also

Other `spca`: [aggregate_by_group\(\)](#), [change_loadings_sign_spca\(\)](#), [change_sign\(\)](#), [change_weights_sign_spca\(\)](#), [compare_spca\(\)](#), [is.spca\(\)](#), [new_spca\(\)](#), [plot.spca\(\)](#), [show_correlations\(\)](#), [show_weights\(\)](#), [spca\(\)](#), [spca_object](#), [summary.spca\(\)](#)

Examples

```
data(holzinger)
ho_cspca = spca(holzinger, n_comps = 4)
ho_cspca
print(ho_cspca, contributions = FALSE, digits = 4)
```

qqplot_sPCA

Wachter QQ Plot for PCA Eigenvalues

Description

Produce a QQ plot comparing the eigenvalues of a fitted PCA with Marchenko–Pastur (Wachter) theoretical quantiles.

Usage

```
qqplot_spca(
  pca_fit,
  n_vars = NULL,
  n_obs = NULL,
  gamma = NULL,
  cor = TRUE,
  common_var = 1,
  n_plot = NULL,
```

```

    n_fitline = NULL,
    addtitle = TRUE,
    show_plot = TRUE,
    return_plot = FALSE
  )

## S3 method for class 'spca_pca'
qqplot_sPCA(
  pca_fit,
  n_vars = NULL,
  n_obs = NULL,
  gamma = NULL,
  cor = TRUE,
  common_var = 1,
  n_plot = NULL,
  n_fitline = NULL,
  addtitle = TRUE,
  show_plot = TRUE,
  return_plot = FALSE
)

## S3 method for class 'spca'
qqplot_sPCA(
  pca_fit,
  n_vars = NULL,
  n_obs = NULL,
  gamma = NULL,
  cor = TRUE,
  common_var = 1,
  n_plot = NULL,
  n_fitline = NULL,
  addtitle = TRUE,
  show_plot = TRUE,
  return_plot = FALSE
)

```

Arguments

<code>pca_fit</code>	An object returned by <code>pca()</code> .
<code>n_vars</code>	An integer scalar or NULL. Number of variables. If NULL, obtain it from the number of rows of <code>pca_fit\$weights</code> .
<code>n_obs</code>	An integer scalar or NULL. Number of observations. If NULL, obtain it from <code>pca_fit\$n_obs</code> .
<code>gamma</code>	A positive numeric scalar or NULL. Aspect ratio. If NULL, use <code>n_obs / n_vars</code> .
<code>cor</code>	A logical scalar retained for compatibility.
<code>common_var</code>	A positive numeric scalar. Common variance used for the Marchenko–Pastur quantiles.

n_plot	An integer scalar or NULL. Number of leading eigenvalues.
n_fitline	An integer scalar or NULL. If positive, fit a line using the last n_fitline points. If negative, exclude the largest abs(n_fitline) points.
addtitle	A logical scalar indicating whether to add a title.
show_plot	A logical scalar indicating whether to print the plot.
return_plot	A logical scalar indicating whether to return the plot.

Value

If return_plot = TRUE, a ggplot object; otherwise NULL invisibly.

See Also

Other pca: [pca\(\)](#), [screeplot_spc\(\)](#), [spca_screplot\(\)](#), [wachter_qqplot\(\)](#)

screeplot_spc	<i>Plot PCA Eigenvalues in a Screeplot</i>
---------------	--

Description

Plot the leading eigenvalues of a fitted PCA against component order.

Usage

```
screeplot_spc(
  pca_fit,
  n_plot = NULL,
  ylab = "eigenvalues",
  addtitle = TRUE,
  show_plot = TRUE,
  return_plot = FALSE
)

## S3 method for class 'spca_pca'
screeplot_spc(
  pca_fit,
  n_plot = NULL,
  ylab = "eigenvalues",
  addtitle = TRUE,
  show_plot = TRUE,
  return_plot = FALSE
)

## S3 method for class 'spca'
screeplot_spc(
  pca_fit,
```

```

    n_plot = NULL,
    ylab = "eigenvalues",
    addtitle = TRUE,
    show_plot = TRUE,
    return_plot = FALSE
  )

```

Arguments

<code>pca_fit</code>	An object returned by <code>pca()</code> .
<code>n_plot</code>	An integer scalar or <code>NULL</code> . Number of leading eigenvalues.
<code>ylab</code>	A character scalar used as the y-axis label.
<code>addtitle</code>	A logical scalar indicating whether to add a title.
<code>show_plot</code>	A logical scalar indicating whether to print the plot.
<code>return_plot</code>	A logical scalar indicating whether to return the plot.

Value

If `return_plot = TRUE`, a `ggplot` object; otherwise `NULL` invisibly.

See Also

Other `pca`: `pca()`, `qqplot_spca()`, `spca_screeplot()`, `wachter_qqplot()`

<code>show_correlations</code>	<i>Show Correlations from an SPCA Object</i>
--------------------------------	--

Description

Print and optionally return the mutual correlations among sparse principal components and their correlations with the corresponding principal components.

Usage

```

show_correlations(spca_obj, ...)

## S3 method for class 'spca'
show_correlations(
  spca_obj,
  type = "both",
  digits = 2,
  print_matrices = TRUE,
  return_matrices = FALSE,
  ...
)

```

Arguments

spca_obj	An object of class spca.
...	Additional arguments reserved for S3 method compatibility.
type	A character value specifying which correlations to show. Values beginning with "s", "p", or "b" select the mutual sPC correlations, the correlations with the corresponding PCs, or both, respectively. The default is "both".
digits	A non-negative integer scalar (default 2). Number of decimal places used for printing. Returned matrices are not rounded.
print_matrices	A logical value (default TRUE). If TRUE, print the requested correlations.
return_matrices	A logical value (default FALSE). If TRUE, return the requested unrounded numeric matrix or matrices.

Value

If return_matrices = TRUE, a numeric matrix when one type of correlation is requested, or a named list of two numeric matrices when type = "both". Otherwise, returns NULL invisibly.

See Also

Other spca: [aggregate_by_group\(\)](#), [change_loadings_sign_spca\(\)](#), [change_sign\(\)](#), [change_weights_sign_spca\(\)](#), [compare_spca\(\)](#), [is.spca\(\)](#), [new_spca\(\)](#), [plot.spca\(\)](#), [print.spca\(\)](#), [show_weights\(\)](#), [spca\(\)](#), [spca_object](#), [summary.spca\(\)](#)

Examples

```
data(holzinger)
ho_cspca = spca(holzinger, n_comps = 3)
show_correlations(ho_cspca)
show_correlations(ho_cspca, type = "s", return_matrices = TRUE)
```

show_weights

Show SPCA Weights or Contributions

Description

Show selected nonzero component weights or their unit-L1 contributions.

Usage

```
show_weights(spca_obj, ...)

## S3 method for class 'spca'
show_weights(
  spca_obj,
```

```

    cols = NULL,
    contribution = TRUE,
    print_list = TRUE,
    return_list = FALSE,
    ...
)

```

Arguments

<code>spca_obj</code>	An object of class <code>spca</code> .
<code>...</code>	Additional arguments reserved for S3 method compatibility.
<code>cols</code>	An integer vector or <code>NULL</code> . Components to show.
<code>contribution</code>	A logical value. If <code>TRUE</code> , show unit-L1 contributions; otherwise, show the original nonzero weights.
<code>print_list</code>	A logical value indicating whether to print the result.
<code>return_list</code>	A logical value indicating whether to return the result.

Value

The selected weights or contributions when requested; otherwise `NULL` invisibly.

See Also

Other `spca`: [aggregate_by_group\(\)](#), [change_loadings_sign_spca\(\)](#), [change_sign\(\)](#), [change_weights_sign_spca\(\)](#), [compare_spca\(\)](#), [is.spca\(\)](#), [new_spca\(\)](#), [plot.spca\(\)](#), [print.spca\(\)](#), [show_correlations\(\)](#), [spca\(\)](#), [spca_object](#), [summary.spca\(\)](#)

`spca`

Compute LS-SPCA Components

Description

Compute least squares sparse principal components (LS-SPCA) from a data matrix or from a covariance/correlation matrix.

Usage

```

spca(
  M,
  n_comps = NULL,
  alpha = 0.95,
  ncomp_by_cvexp = NULL,
  method = c("cspca", "uspca", "pspca"),
  var_selection = c("fwd", "bkw", "step"),
  objective = c("r2", "cvexp"),
  intensive = FALSE,

```

```

fat_matrix = NULL,
fixed_index_list = NULL,
center_data = FALSE,
scale_data = FALSE,
pm_weights = FALSE,
eps_pm_weights = 1e-04,
maxiter_pm_weights = 1000,
pm_varsel = FALSE,
eps_pm_varsel = 1e-04,
maxiter_pm_varsel = 500
)

```

Arguments

M	A numeric matrix or data frame. If M is square, it is treated as a covariance/correlation matrix and the tall backend is used. Otherwise, M is treated as an $n \times p$ data matrix.
n_comps	A nonnegative integer scalar or NULL (default NULL). Number of components to compute. If NULL, ncomp_by_cvexp is used to determine the number of components. At least one of n_comps and ncomp_by_cvexp must be supplied.
alpha	A numeric scalar in (0, 1] (default 0.95). Target retained proportion used by variable selection.
ncomp_by_cvexp	A numeric scalar in (0, 1] or NULL (default NULL). If n_comps = NULL, components are computed until cumulative variance explained reaches this value.
method	A character vector (default first element "cspca"). LS-SPCA variant. Accepted values are "cspca", "uspc", and "pspc"; only the first letter is used.
var_selection	A character vector (default first element "fwd"). Variable-selection algorithm. Values starting with "f" use forward selection, values starting with "b" use backward elimination, and values starting with "s" use forward-stepwise selection.
objective	A character vector (default first element "r2"). Stopping criterion for variable selection. Values starting with "r" use the squared-correlation criterion; values starting with "c" use cumulative variance explained.
intensive	A logical value (default FALSE). If TRUE, the tall backend uses intensive forward CVEXP selection. This option is not available for fat matrices.
fat_matrix	A logical value or NULL (default NULL). If NULL, data matrices with more columns than rows use the fat backend, and all other inputs use the tall backend. If TRUE, the fat backend is requested. If FALSE, the tall backend is used.
fixed_index_list	A list of integer-valued vectors, a factor, or NULL (default NULL). If supplied, it must define a mutually exclusive and exhaustive partition of the variables with at least two groups. List indices are 1-based.
center_data	A logical value (default FALSE). If TRUE, center data-matrix columns before fitting. Ignored when M is treated as a covariance/correlation matrix.
scale_data	A logical value (default FALSE). If TRUE, scale data-matrix columns before fitting. Ignored when M is treated as a covariance/correlation matrix.

<code>pm_weights</code>	A logical value (default FALSE). If TRUE, use the power method for PC and sparse-weight eigenvectors.
<code>eps_pm_weights</code>	A positive numeric scalar (default 1e-4). Convergence tolerance for <code>pm_weights</code> .
<code>maxiter_pm_weights</code>	A positive integer scalar (default 1000). Maximum number of iterations for <code>pm_weights</code> .
<code>pm_varsel</code>	A logical value (default FALSE). If TRUE, use the power method inside variable selection.
<code>eps_pm_varsel</code>	A positive numeric scalar (default 1e-4). Convergence tolerance for <code>pm_varsel</code> .
<code>maxiter_pm_varsel</code>	A positive integer scalar (default 500). Maximum number of iterations for <code>pm_varsel</code> .

Details

Data matrices are routed to the tall or fat C++ backend. Square matrices are treated as covariance/correlation matrices and use the tall backend.

Variable selection is controlled by `var_selection`, `objective`, and `intensive`.

<code>var_selection</code>	<code>objective</code>	Algorithm
"fwd"	"r2"	Forward selection with squared-correlation stopping
"bkw"	"r2"	Backward elimination with squared-correlation stopping
"step"	"r2"	Forward-stepwise selection with squared-correlation stopping
"fwd"	"cvexp"	Forward selection with CVEXP stopping
"bkw"	"cvexp"	Backward elimination with CVEXP stopping
"step"	"cvexp"	Forward-stepwise selection with CVEXP stopping
<code>intensive = TRUE</code> requires "fwd"	"cvexp"	Intensive forward CVEXP selection

The fat backend currently supports regression-based forward variable selection only: `var_selection = "f"` and `intensive = FALSE`. Other combinations generate an error.

The returned object is documented in [spca_object](#).

Value

An object of class `spca`.

See Also

Other `spca`: [aggregate_by_group\(\)](#), [change_loadings_sign_spca\(\)](#), [change_sign\(\)](#), [change_weights_sign_spca\(\)](#), [compare_spca\(\)](#), [is.spca\(\)](#), [new_spca\(\)](#), [plot.spca\(\)](#), [print.spca\(\)](#), [show_correlations\(\)](#), [show_weights\(\)](#), [spca_object](#), [summary.spca\(\)](#)

Examples

```
data(holzinger)
#default
ho_cspca = spca(holzinger, n_comps = 4)
#uncorrelated components and subsets determined using CVEXP as stopping rule
ho_uspca = spca(holzinger, n_comps = 4, method = "uspca",
               objective = "cvexp")
```

spca_object

Sparse Principal Component Analysis Object

Description

Objects of class `spca` are returned by the fitting functions `spca()`, `pca()`, and `new_spca()`. Objects returned by `pca()` also inherit from class `spca_pca`.

Components

An object of class `spca` is a list with the following elements:

weights $p \times r$ matrix of sparse weights.

contributions $p \times r$ matrix of weights scaled to unit L_1 norm within each sPC.

n_comps Number of sPCs.

cardinality Number of nonzero weights in each sPC.

vexp Variance explained by each sPC.

vexp_pc Variance explained by the corresponding PCs.

cvexp Cumulative variance explained by the sPCs.

rvexp Ratio of vexp to the variance explained by the corresponding PC.

rcvexp Ratio of cvexp to the cumulative variance explained by the corresponding PCs.

cor_with_pc Correlation between each sPC and the corresponding PC.

tot_var Total variance of the data.

weights_list List of nonzero weight vectors, one per sPC.

spc_cor $n_{comps} \times n_{comps}$ correlation matrix of the sPC scores.

indices List of variable indices with nonzero weights, one per sPC.

scores Optional matrix of sPC scores, returned only when a data matrix is supplied.

parameters List of parameters used to compute an `spca()` fit.

call Matched call used to compute an `spca()` fit.

eigenvalues For `pca()` objects, the available PCA eigenvalues.

n_obs For `pca()` objects, the number of observations when available.

method_name For `new_spca()` objects, an optional method label.

For backward compatibility, methods also accept objects from earlier package versions containing `loadings` and `loadings_list` instead of `weights` and `weights_list`.

See Also

Other spca: [aggregate_by_group\(\)](#), [change_loadings_sign_spca\(\)](#), [change_sign\(\)](#), [change_weights_sign_spca\(\)](#), [compare_spca\(\)](#), [is.spca\(\)](#), [new_spca\(\)](#), [plot.spca\(\)](#), [print.spca\(\)](#), [show_correlations\(\)](#), [show_weights\(\)](#), [spca\(\)](#), [summary.spca\(\)](#)

spca_screepLOT

Plot Eigenvalues in a Scree Plot (Deprecated)

Description

`spca_screepLOT()` is retained for backward compatibility. Use [screepLOT_spca\(\)](#) with objects returned by [pca\(\)](#) in new code.

Usage

```
spca_screepLOT(
  eigenvalues,
  n_plot = NULL,
  ylab = "eigenvalues",
  addtitle = TRUE,
  show_plot = TRUE,
  return_plot = FALSE
)
```

Arguments

<code>eigenvalues</code>	A numeric vector of eigenvalues, or an object returned by pca() .
<code>n_plot</code>	An integer scalar or NULL. Number of leading eigenvalues.
<code>ylab</code>	A character scalar used as the y-axis label.
<code>addtitle</code>	A logical scalar indicating whether to add a title.
<code>show_plot</code>	A logical scalar indicating whether to print the plot.
<code>return_plot</code>	A logical scalar indicating whether to return the plot.

Value

If `return_plot = TRUE`, a ggplot object; otherwise NULL invisibly.

See Also

Other pca: [pca\(\)](#), [qqplot_spca\(\)](#), [screepLOT_spca\(\)](#), [wachter_qqplot\(\)](#)

summary.spca	<i>Summarize an spca Object</i>
--------------	---------------------------------

Description

Print and optionally return summary statistics for evaluating an spca object and comparing it with the corresponding PCA solution.

Usage

```
## S3 method for class 'spca'
summary(
  object,
  cols,
  contributions = TRUE,
  variance_metrics = c("both", "cumulative_relative", "relative", "none"),
  min_weight = FALSE,
  cor_with_pc = TRUE,
  return_table = FALSE,
  print_table = TRUE,
  thresh_card = 1e-08,
  ...
)
```

Arguments

object	An object of class spca.
cols	An integer vector of component indices. If missing, all available components are included. If a single integer is supplied, components 1:cols are included.
contributions	A logical value (default TRUE). If TRUE, minimum nonzero values are computed from percentage contributions; otherwise, they are computed from weights.
variance_metrics	A character vector (default first element "both"). Controls which relative variance metrics are included. Accepted values are "relative", "cumulative_relative", "both", and "none".
min_weight	A logical value (default FALSE). If TRUE, include the minimum nonzero weight or contribution.
cor_with_pc	A logical value (default FALSE). If TRUE, include correlations between sPCs and the corresponding PCs when available.
return_table	A logical value (default FALSE). If TRUE, return the raw numeric summary matrix.
print_table	A logical value (default TRUE). If TRUE, print the formatted summary table.
thresh_card	A numeric scalar (default 1e-8). Values with absolute magnitude at or below this threshold are treated as zero when computing cardinality.
...	Further arguments. These are currently unused and trigger an error if supplied.

Details

For each component, the following summaries can be computed:

Vexp	The percentage variance explained.
Cvexp	The percentage cumulative variance explained.
Rvexp	The variance explained relative to the corresponding PC.
Rcvexp	The cumulative variance explained relative to the corresponding PCs.
Card	The cardinality, that is the number of non zero weights.
Min weight/Min cont	The minimum absolute value of the nonzero weights or contributions, if requested.
r	The correlation between sPCs and the corresponding PCs, if requested.

Value

If `return_table = TRUE`, returns a numeric matrix with the selected summary statistics. Otherwise, returns NULL invisibly.

See Also

Examples in [aggregate_by_group](#).

Other spca: [aggregate_by_group\(\)](#), [change_loadings_sign_spca\(\)](#), [change_sign\(\)](#), [change_weights_sign_spca\(\)](#), [compare_spca\(\)](#), [is.spca\(\)](#), [new_spca\(\)](#), [plot.spca\(\)](#), [print.spca\(\)](#), [show_correlations\(\)](#), [show_weights\(\)](#), [spca\(\)](#), [spca_object](#)

Examples

```
data(holzinger)
ho_cspca = spca(holzinger, n_comps = 2)
summary(ho_cspca)
```

wachter_qqplot

Wachter QQ Plot for Eigenvalues (Deprecated)

Description

`wachter_qqplot()` is retained for backward compatibility. Use [qqplot_spca\(\)](#) with objects returned by [pca\(\)](#) in new code.

Usage

```
wachter_qqplot(
  eigenvalues,
  p = NULL,
  n,
  gamma,
  cor = TRUE,
```

```

    common_var = 1,
    n_plot = NULL,
    n_fitline = NULL,
    addtitle = TRUE,
    show_plot = TRUE,
    return_plot = FALSE
  )

```

Arguments

eigenvalues	A numeric vector of eigenvalues in decreasing order, or an object returned by pca() .
p	An integer scalar or NULL. Number of variables.
n	An integer scalar. Number of observations.
gamma	A positive numeric scalar. Aspect ratio. If omitted, use n / p .
cor	A logical scalar retained for compatibility.
common_var	A positive numeric scalar. Common variance used for the Marchenko–Pastur quantiles.
n_plot	An integer scalar or NULL. Number of leading eigenvalues.
n_fitline	An integer scalar or NULL. If positive, fit a line using the last <code>n_fitline</code> points. If negative, exclude the largest <code>abs(n_fitline)</code> points.
addtitle	A logical scalar indicating whether to add a title.
show_plot	A logical scalar indicating whether to print the plot.
return_plot	A logical scalar indicating whether to return the plot.

Value

If `return_plot = TRUE`, a ggplot object; otherwise NULL invisibly.

See Also

Other pca: [pca\(\)](#), [qqplot_spca\(\)](#), [screeplot_spca\(\)](#), [spca_screeplot\(\)](#)

Index

- * **datasets**
 - holzinger, 9
 - holzinger_scales, 10
- * **pca**
 - pca, 12
 - qqplot_spca, 17
 - screeplot_spca, 19
 - spca_screeplot, 26
 - wachter_qqplot, 28
- * **spca**
 - aggregate_by_group, 4
 - change_loadings_sign_spca, 5
 - change_sign, 5
 - change_weights_sign_spca, 6
 - compare_spca, 7
 - is.spca, 10
 - new_spca, 11
 - plot.spca, 13
 - print.spca, 16
 - show_correlations, 20
 - show_weights, 21
 - spca, 22
 - spca_object, 25
 - summary.spca, 27
- aggregate_by_group, 4, 5, 6, 8, 10, 11, 15, 17, 21, 22, 24, 26, 28
- aggregate_by_group(), 3
- change_loadings_sign_spca, 4, 5, 6, 8, 10, 11, 15, 17, 21, 22, 24, 26, 28
- change_loadings_sign_spca(), 3
- change_sign, 4, 5, 5, 6, 8, 10, 11, 15, 17, 21, 22, 24, 26, 28
- change_sign(), 3, 5, 6
- change_weights_sign_spca, 4–6, 6, 8, 10, 11, 15, 17, 21, 22, 24, 26, 28
- change_weights_sign_spca(), 3
- compare_spca, 4–6, 7, 10, 11, 15, 17, 21, 22, 24, 26, 28
- compare_spca(), 3
- holzinger, 9
- holzinger_scales, 10
- is.spca, 4–6, 8, 10, 11, 15, 17, 21, 22, 24, 26, 28
- is.spca(), 3
- new_spca, 4–6, 8, 10, 11, 15, 17, 21, 22, 24, 26, 28
- new_spca(), 3, 25
- pca, 12, 19, 20, 26, 29
- pca(), 3, 18, 20, 25, 26, 28, 29
- plot(), 3
- plot.spca, 4–6, 8, 10, 11, 13, 17, 21, 22, 24, 26, 28
- print(), 3
- print.spca, 4–6, 8, 10, 11, 15, 16, 21, 22, 24, 26, 28
- qqplot_spca, 12, 13, 17, 20, 26, 29
- qqplot_spca(), 3, 28
- screeplot_spca, 13, 19, 19, 26, 29
- screeplot_spca(), 3, 26
- show_correlations, 4–6, 8, 10, 11, 15, 17, 20, 22, 24, 26, 28
- show_weights, 4–6, 8, 10, 11, 15, 17, 21, 21, 24, 26, 28
- show_weights(), 3
- spca, 4–6, 8, 10, 11, 15, 17, 21, 22, 22, 26, 28
- spca(), 2, 25
- spca-package, 2
- spca_object, 2–6, 8, 10, 11, 13, 15, 17, 21, 22, 24, 25, 28
- spca_screeplot, 13, 19, 20, 26, 29
- spca_screeplot(), 3
- summary(), 3

`summary.sPCA`, [4–8](#), [10](#), [11](#), [15](#), [17](#), [21](#), [22](#), [24](#),
[26](#), [27](#)

`wachter.qqplot`, [13](#), [19](#), [20](#), [26](#), [28](#)

`wachter.qqplot()`, [3](#)