Package: spca (via r-universe)

October 30, 2026

Type Package

Title Least Squares Sparse Principal Components Analysis

Version 1.1.5

Date 2026-09-28

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> and Merola and Chen (2019) <doi: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

VignetteBuilder knitr

Config/testthat/edition 3

LinkingTo Rcpp, RcppEigen

RoxygenNote 7.3.2

Roxygen list(markdown = TRUE)

LazyData true

NeedsCompilation yes

Repository https://merolagio.r-universe.dev

Date/Publication 2026-09-30 12:04:38 UTC

RemoteUrl https://github.com/merolagio/spca

RemoteRef HEAD

RemoteSha b4f4d4f3b2366ee972c504c74ed6a9cb80e9a13e

Contents

spca-package . . . 2
aggregate_by_group . . . 4
change_sign . . . 5
compare_spca . . . 6
holzinger . . . 8
holzinger_scales . . . 9
is.pca . . . 10
is.spca . . . 10
mp_qqplot . . . 11
new_spca . . . 13
pca . . . 14
plot.pca . . . 16
print.pca . . . 19
scree_plot . . . 20
show_correlations . . . 22
show_weights . . . 23
spca . . . 25
spca_object . . . 27
summary.pca . . . 29

Index **31**

`spca-package` *Least Squares Sparse Principal Components Analysis*

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.
- `scree_plot()` and `mp_qqplot()` provide diagnostic plots for objects returned by `pca()`.

Utilities

- `is.spca()` Verifies whether an object is a proper `spca` object.
- `is.pca()` Verifies whether an object is a proper `pca` 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 *Aggregate Weights or Contributions of a* pca *or* spca *Object by Group*

Description

Aggregate component weights or contributions according to a grouping variable.

Usage

```
aggregate_by_group(
  object,
  variable_groups,
  cols = NULL,
  only_nonzero = TRUE,
  contributions = TRUE,
  digits = ifelse(contributions, 1, 3),
  print_table = TRUE,
  return_table = FALSE,
  spca_obj = NULL
)

## S3 method for class 'spca'
aggregate_by_group(
  object,
  variable_groups,
  cols = NULL,
  only_nonzero = TRUE,
  contributions = TRUE,
  digits = ifelse(contributions, 1, 3),
  print_table = TRUE,
  return_table = FALSE,
  spca_obj = NULL
)

## S3 method for class 'pca'
aggregate_by_group(object, ...)
```

Arguments

object	An object of class pca or spca.
variable_groups	A vector or factor with one group label per variable.
cols	An integer vector selecting components. If a single positive integer is supplied, components 1:cols are included. If NULL, all components are included.
only_nonzero	A logical value indicating whether to omit variable_groups whose values are zero in every selected component.

contributions	A logical value. If TRUE, aggregate percentage contributions; otherwise, aggregate weights.
digits	Number of digits used in printed output.
print_table	A logical value indicating whether to print the table.
return_table	A logical value indicating whether to return the table visibly.
spca_obj	Deprecated alias for object (default NULL). Supply only one of object and spca_obj. Using the old argument name issues a warning.
...	Additional arguments passed to aggregate_by_group.spca.

Methods (by class)

- aggregate_by_group(pca): Aggregate a pca object by group.

See Also

Other spca: change_loadings_sign_spca(), change_sign(), change_weights_sign_spca(), compare_spca(), is.spca(), new_spca(), plot.pca(), print.pca(), show_contributions_spca(), show_correlations(), show_weights(), spca(), spca_object, summary.pca()

Other pca: change_sign(), is.pca(), mp_qqplot(), pca(), print.pca(), scree_plot(), show_correlations(), show_weights(), spca_screeplot(), wachter_qqplot()

---

change_sign *Change Component Signs in a* pca *or* spca *Object*

Description

Change the signs of selected Weights in a fitted object.

Usage

```
change_sign(object, index_to_change, spca_obj = NULL)

## S3 method for class 'spca'
change_sign(object, index_to_change, spca_obj = NULL)

## S3 method for class 'pca'
change_sign(object, ...)
```

Arguments

object	An object of class pca or spca.
index_to_change	An integer vector of component indices whose signs should be changed.
spca_obj	Deprecated alias for object (default NULL). Supply only one of object and spca_obj. Using the old argument name issues a warning.
...	Additional arguments passed to change_sign.spca.

Value

The modified object, preserving its classes.

Methods (by class)

- `change_sign(pca)`: Change component signs in a pca object.

See Also

Other spca: `aggregate_by_group()`, `change_loadings_sign_spca()`, `change_weights_sign_spca()`, `compare_spca()`, `is.spca()`, `new_spca()`, `plot.pca()`, `print.pca()`, `show_contributions_spca()`, `show_correlations()`, `show_weights()`, `spca()`, `spca_object`, `summary.pca()`

Other pca: `aggregate_by_group()`, `is.pca()`, `mp_qqplot()`, `pca()`, `print.pca()`, `scree_plot()`, `show_correlations()`, `show_weights()`, `spca_screeplot()`, `wachter_qqplot()`

Examples

```
data(holzinger)
ho_cspca = spca(holzinger, n_comps = 4)
show_correlations(ho_cspca)
# In applications we would change only the fourth set of weights
ho_changed = change_sign(ho_cspca, index_to_change = c(2, 4))
is.spca(ho_changed)
show_weights(ho_cspca, cols = 2)
show_weights(ho_changed, cols = 2)
show_correlations(ho_changed)
```

---

compare_spca *Compare Two or More* spca *Solutions*

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 = "black",
  color_scale = c("ggplot", "cbb", "printsafe", "bw"),
  col_short_names = TRUE,
  print_tables = TRUE,
  print_weights = FALSE,
  print_summary = TRUE,
  show_plot = TRUE,
  return_tables = FALSE,
  return_plot = FALSE
)
```

Arguments

obj_list	A list of two or more spca objects.
n_comps	An integer scalar or NULL (default NULL). Number of components to compare. If NULL, the minimum number of available components across objects is used.
contributions	A logical value (default TRUE). If TRUE, compare percentage contributions; otherwise, compare weights.
only_nonzero	A logical value (default TRUE). If TRUE, only variables with at least one nonzero weight or contribution are plotted or printed.
variable_groups	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.
plot_weights	A logical value (default TRUE). If TRUE, plot the weights or contributions.
plot_type	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 "black"). 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 all table printing. Takes priority over print_weights and print_summary.
print_weights	A logical value (default FALSE). If TRUE, print the weights or contributions table.
print_summary	A logical value (default TRUE). If TRUE, print the summary statistics table when print_tables = TRUE.
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.pca()`, `print.pca()`, `show_contributions_spca()`, `show_correlations()`, `show_weights()`, `spca()`, `spca_object`, `summary.pca()`

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` *Holzinger–Swineford Student Ability Scales*

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.pca *Test for PCA Objects*

Description

Check whether an object has class pca and contains the core elements produced by pca().

Usage

```
is.pca(x)
```

Arguments

x	An object to test.

Details

Performs a lightweight check of the class and the presence of core elements. It does not validate their values or dimensions. Scores and the number of observations are not required because they may be unavailable for covariance-matrix input.

Value

A logical value: TRUE if the required class and core elements are present, and FALSE otherwise.

See Also

Other pca: aggregate_by_group(), change_sign(), mp_qqplot(), pca(), print.pca(), scree_plot(), show_correlations(), show_weights(), spca_screeplot(), wachter_qqplot()

Examples

```
data(holzinger)
ho_pca = pca(holzinger, n_comps = 2, screeplot = FALSE, qq_plot = FALSE)
is.pca(ho_pca)
```

is.spca *Test for SPCA Objects*

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.pca()`, `print.pca()`, `show_contributions_spca()`, `show_correlations()`, `show_weights()`, `spca()`, `spca_object`, `summary.pca()`

Examples

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

---

mp_qqplot *PCA Diagnostic qqplot*

Description

Produce a qq-plot comparing the eigenvalues of a fitted PCA with Marchenko–Pastur theoretical quantiles.

Usage

```
mp_qqplot(
  pca_fit,
  n_vars = NULL,
  n_obs = NULL,
  gamma = NULL,
  cor = TRUE,
  common_var = NULL,
  n_plot = NULL,
  n_fitline = NULL,
  addtitle = TRUE,
```

```
  show_plot = TRUE,
  return_plot = FALSE
)

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

Arguments

pca_fit	An object of class "pca", as returned by pca().
n_vars	An integer scalar or NULL (default NULL). Number of variables. If NULL, obtain it from the number of rows of pca_fit$weights.
n_obs	An integer scalar or NULL (default NULL). Number of observations. If NULL, obtain it from pca_fit$n_obs.
gamma	A positive numeric scalar or NULL (default NULL). Ratio of observations to variables. If NULL, use n_obs / n_vars.
cor	A logical scalar (default TRUE). If TRUE, use the correlation-matrix reference: set common_var = 1 and scale the theoretical quantiles to sum to n_vars. If FALSE, retain the covariance scale specified by common_var.
common_var	A positive numeric scalar or NULL (default NULL). Assumed common population variance for the Marchenko–Pastur quantiles. Required when cor = FALSE; set to 1 when cor = TRUE.
n_plot	An integer scalar or NULL (default NULL). Number of leading eigenvalues. If NULL, use all stored or supplied eigenvalues.
n_fitline	An integer scalar or NULL (default NULL: no line). Zero also omits the line. If positive, fit a line using the last n_fitline points. If negative, exclude the largest abs(n_fitline) points.
addtitle	A logical scalar (default TRUE). Add a title.
show_plot	A logical scalar (default TRUE). Print the plot.
return_plot	A logical scalar (default FALSE). Return the plot.

Details

This diagnostic relies on Marchenko–Pastur theory for the null eigenvalue distribution of a (possibly high-dimensional) sample covariance matrix, and is therefore only meaningful for objects of class

"pca" (i.e. objects returned by pca()). It does not apply to sparse fits returned by spca() (class "spca"), for which the reference distribution does not hold; no method is defined for that class, so calling it on such an object raises the standard "no applicable method" error rather than producing a plot.

The Marchenko–Pastur distribution depends on the ratio of variables to observations. The gamma argument uses its reciprocal, n_obs / n_vars, as required by RMTstat::qmp. Both counts must be available, even when gamma is supplied. The covariance reference assumes independent variables with a common population variance. With cor = TRUE, the theoretical quantiles are scaled to sum to n_vars. A warning is issued if the stored eigenvalues do not sum to n_vars within a tolerance of 1e-8 * n_vars. This check assumes that all nonzero eigenvalues are stored; it does not establish that the original matrix is a correlation matrix. With cor = FALSE, the quantiles retain the scale set by common_var. A fitted line must use at least two of the plotted points.

Value

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

See Also

Other pca: aggregate_by_group(), change_sign(), is.pca(), pca(), print.pca(), scree_plot(), show_correlations(), show_weights(), spca_screeplot(), wachter_qqplot()

Examples

```
data(holzinger)
ho_pca = pca(holzinger)
# from the screeplot we may choose to retain 4 components
# produce a Wachter qqplot fitting a line to all but the largest
# 4 eigenvalues.
# Since pca was fitted using the data matrix, the fit carries the number
# of observations. Otherwise we would need to pass `n_obs` .
mp_qqplot(ho_pca, n_fitline = -4)
# The qq-plot indicates that the 4th eigenvalue is compatible
# with that of a random matrix.
```

---

new_spca *Construct an SPCA Object from a Set of Weights*

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.pca(), print.pca(), show_contributions_spca(), show_correlations(), show_weights(), spca(), spca_object, summary.pca()

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 = TRUE,
  qq_plot = FALSE,
  n_obs = NULL,
  neigen_toplot = NULL,
  cor = TRUE,
  common_var = NULL,
  pm = FALSE,
  eps_pm = 1e-04,
  maxiter_pm = 1000
)
```

Arguments

M	A data matrix, correlation matrix, or covariance matrix.
n_comps	An integer scalar or NULL (default NULL). Number of components to retain. If NULL, all components are retained up to the maximum allowed by the selected backend.
center_data	A logical value (default FALSE). If TRUE, center variables to zero mean. If M is detected as a data matrix and any column mean is nonzero, centering is performed automatically.
scale_data	A logical value (default FALSE). If TRUE, scale variables.
fat_matrix	A logical value or NULL (default NULL). If NULL, the backend is selected automatically: data matrices with $n < p$ use the fat backend and all other inputs use the tall backend. If TRUE, request the fat backend. If FALSE, use the tall backend. Covariance and correlation matrices always use the tall backend.
screeplot	A logical value (default TRUE). If TRUE, produce a scree plot.
qq_plot	A logical value (default FALSE). If TRUE, produce a Wachter qq-plot with mp_qqplot.
n_obs	An integer scalar or NULL (default NULL). Number of rows in the original data set. Required when qq_plot = TRUE and M 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); the diagnostic plot uses common_var for the Marchenko–Pastur quantiles.
common_var	A positive numeric scalar or NULL (default NULL). Common population variance assumed for the Wachter qq-plot. Required when qq_plot = TRUE and cor = FALSE. When cor = TRUE, the reference variance is set to 1.
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, so passing this saves its computation from the data matrix. If a covariance or correlation matrix is passed, the scores cannot be computed and are returned as NULL. The fat backend computes PCA in row space and converts the retained eigenvectors back to variable weights.

Value

An pca_object which is the same as 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: aggregate_by_group(), change_sign(), is.pca(), mp_qqplot(), print.pca(), scree_plot(), show_correlations(), show_weights(), spca_screeplot(), wachter_qqplot()

Examples

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

---

plot.pca *Plot the Weights or Contributions of a* pca *or* spca *Object*

Description

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

Usage

```
## S3 method for class 'pca'
plot(x, ...)

## 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 =
  c("none", "auto"), 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

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

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 "auto" or "none" (default "none"). If "auto", row names of the weight matrix are used as variable names; if row names are missing, no variable names are printed. NULL is a deprecated alias for "auto" retained for backward compatibility; using it emits a warning to use "auto" instead. 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.

Functions

- plot(pca): A pca object is plotted identically to an spca object.

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.pca(), show_contributions_spca(), show_correlations(), show_weights(), spca(), spca_object, summary.pca()

Other spca: aggregate_by_group(), change_loadings_sign_spca(), change_sign(), change_weights_sign_spca(), compare_spca(), is.spca(), new_spca(), print.pca(), show_contributions_spca(), show_correlations(), show_weights(), spca(), spca_object, summary.pca()

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.pca` *Print the Weights or Contributions of a* `pca` *or* `spca` *Object*

Description

Print sparse weights, or the corresponding percentage contributions, from a fitted 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 'pca'
print(x, ...)

## 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

`x`	An object of class `spca` or `pca`.
`...`	Further arguments. These are currently unused and trigger an error if supplied.
`cols`	An integer vector or `NULL` (default `NULL`). Components to print. If `NULL`, all components are printed. If a single integer is supplied, components `1:cols` are printed.

only_nonzero	A logical value (default TRUE). If TRUE, print only variables with at least one weight or contribution whose absolute value is greater than or equal to thresh_card.
contributions	A logical value (default TRUE). If TRUE, print weights scaled to unit L_1 norm as percentage contributions; otherwise, print L2 unit weights.
digits	An integer scalar (default 3). Number of decimal places used when printing weights. Contributions are printed as percentages with one decimal place.
thresh_card	A numeric scalar (default 1e-07). Values with absolute magnitude below this threshold are treated as zero in the printed table.
return_table	A logical value (default FALSE). If TRUE, return the formatted character matrix.
component_names	A character vector or NULL (default NULL). Optional component names. If NULL, existing column names are used when available; otherwise default names are assigned.

Value

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

Functions

- print(pca): A pca object is printed identically to an spca object.

See Also

Other spca: aggregate_by_group(), change_loadings_sign_spca(), change_sign(), change_weights_sign_spca(), compare_spca(), is.spca(), new_spca(), plot.pca(), show_contributions_spca(), show_correlations(), show_weights(), spca(), spca_object, summary.pca()

Other pca: aggregate_by_group(), change_sign(), is.pca(), mp_qqplot(), pca(), scree_plot(), show_correlations(), show_weights(), spca_screeplot(), wachter_qqplot()

Examples

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

---

scree_plot *S3 Method for Plotting PCA Eigenvalues in a Screeplot*

Description

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

Usage

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

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

Arguments

pca_fit	An object of class "pca", as returned by pca().
n_plot	An integer scalar or NULL (default NULL). Number of leading eigenvalues. If NULL, use all stored or supplied eigenvalues.
ylab	A character scalar (default "eigenvalues"). Y-axis label.
addtitle	A logical scalar (default TRUE). Add a title.
show_plot	A logical scalar (default TRUE). Print the plot.
return_plot	A logical scalar (default FALSE). Return the plot.

Details

This diagnostic applies only to objects of class "pca" (i.e. objects returned by pca()). It does not apply to sparse fits returned by spca() (class "spca"); no method is defined for that class, so calling it on such an object raises the standard "no applicable method" error rather than producing a plot.

Value

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

See Also

Other pca: aggregate_by_group(), change_sign(), is.pca(), mp_qqplot(), pca(), print.pca(), show_correlations(), show_weights(), spca_screeplot(), wachter_qqplot()

Examples

```
data(holzinger)
ho_pca = pca(holzinger, screeplot = FALSE)
# the screeplot can be produced from `pca` directly.
# It can be customized by saving it as a `ggplot` object
myscreeplot = scree_plot(ho_pca, return_plot = TRUE)
myscreeplot  + ggplot2::geom_point(color = "red")
```

---

show_correlations *Show Component Correlations for a* pca *or* spca *Object*

Description

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

Usage

```
show_correlations(
  object,
  type = c("both", "spcs", "pcs"),
  digits = 2,
  print_matrices = TRUE,
  return_matrices = FALSE,
  spca_obj = NULL
)

## S3 method for class 'spca'
show_correlations(
  object,
  type = c("both", "spcs", "pcs"),
  digits = 2,
  print_matrices = TRUE,
  return_matrices = FALSE,
  spca_obj = NULL
)
```

Arguments

object	An object of class pca or spca.
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.

spca_obj Deprecated alias for object (default NULL). Supply only one of object and spca_obj. Using the old argument name issues a warning.

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.pca(), print.pca(), show_contributions_spca(), show_weights(), spca(), spca_object, summary.pca()

Other pca: aggregate_by_group(), change_sign(), is.pca(), mp_qqplot(), pca(), print.pca(), scree_plot(), show_weights(), spca_screeplot(), wachter_qqplot()

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 the Weights or Contributions of a* pca *or* spca *Object*

Description

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

Usage

```
show_weights(
  object,
  cols = NULL,
  contribution = TRUE,
  print_list = TRUE,
  return_list = FALSE,
  ...,
  spca_obj = NULL
)
```

```
## S3 method for class 'spca'
show_weights(
  object,
  cols = NULL,
  contribution = TRUE,
  print_list = TRUE,
  return_list = FALSE,
  ...,
  spca_obj = NULL
)

## S3 method for class 'pca'
show_weights(object, ...)
```

Arguments

object	An object of class pca or spca.
cols	An integer vector selecting components. If a single positive integer is supplied, components 1:cols are included. If NULL, all components are included.
contribution	A logical value. If TRUE, show unit-L1 contributions; otherwise, show the original nonzero weights.
print_list	A logical value indicating whether to print the result.
return_list	A logical value indicating whether to return the result.
...	Additional arguments reserved for S3 method compatibility.
spca_obj	Deprecated alias for object (default NULL). Supply only one of object and spca_obj. Using the old argument name issues a warning.

Value

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

Methods (by class)

- show_weights(pca): The weights or contributions of a pca object are shown identically those of an spca object.

See Also

Other spca: aggregate_by_group(), change_loadings_sign_spca(), change_sign(), change_weights_sign_spca(), compare_spca(), is.spca(), new_spca(), plot.pca(), print.pca(), show_contributions_spca(), show_correlations(), spca(), spca_object, summary.pca()

Other pca: aggregate_by_group(), change_sign(), is.pca(), mp_qqplot(), pca(), print.pca(), scree_plot(), show_correlations(), spca_screeplot(), wachter_qqplot()

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("cvexp", "r2"),
  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", "uspca", and "pspca"; 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 `"cvexp"` for all components). 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`). Each list element gives the fixed variable indices for the corresponding component. A `NULL` or empty element requests variable selection for that component. A list of length one fixes only the first component; to fix a later component, include `NULL` or empty placeholders for the preceding components. Index sets may overlap and need not include every variable. Indices are 1-based. A factor remains available as a shorthand for assigning variables to component-specific groups. Duplicate indices within a component are removed with a warning. If the list is longer than the number of components, extra elements are ignored with a warning.

`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.

`pm_weights` A logical value (default `FALSE`). If `TRUE`, use the power method for PC and sparse-weight eigenvectors.

`eps_pm_weights` A positive numeric scalar (default `1e-4`). Convergence tolerance for `pm_weights`.

`maxiter_pm_weights`
A positive integer scalar (default `1000`). Maximum number of iterations for `pm_weights`.

`pm_varsel` A logical value (default `FALSE`). If `TRUE`, use the power method inside variable selection.

`eps_pm_varsel` A positive numeric scalar (default `1e-4`). Convergence tolerance for `pm_varsel`.

`maxiter_pm_varsel`
A positive integer scalar (default `500`). Maximum number of iterations for `pm_varsel`.

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`.

var_selection	objective	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
intensive = TRUE 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.

If both n_comps and ncomp_by_cvexp are supplied, n_comps takes precedence and automatic stopping by cumulative variance explained is disabled.

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.pca(), print.pca(), show_contributions_spca(), show_correlations(), show_weights(), spca_object, summary.pca()

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")
#fix indices for components 1 and 3 and select them for component 2
ho_fixed = spca(holzinger, n_comps = 3,
                fixed_index_list = list(c(1, 2), NULL, c(2, 4, 6)))
```

---

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.pca(), print.pca(), show_contributions_spca(), show_correlations(), show_weights(), spca(), summary.pca()

summary.pca *Summarize the Variance Explained and Sparsity of a* pca *or* spca *Object*

Description

Print and, optionally, return summary statistics for evaluating an spca object and comparing it with the corresponding PCA solutions. It can also print the summaries of a pca object.

Usage

```
## S3 method for class 'pca'
summary(object, ...)

## S3 method for class 'spca'
summary(
  object,
  cols = NULL,
  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 or pca.
...	Further arguments. These are currently unused and trigger an error if supplied.
cols	An integer vector of component indices or NULL. If NULL (the default), 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 TRUE). 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.

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.

Functions

- summary(pca): Summarize a pca object using the shared summary method.

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.pca(), print.pca(), show_contributions_spca(), show_correlations(), show_weights(), spca(), spca_object

Examples

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

Index

∗ **datasets**
holzinger, 8
holzinger_scales, 9
∗ **pca**
aggregate_by_group, 4
change_sign, 5
is.pca, 10
mp_qqplot, 11
pca, 14
print.pca, 19
scree_plot, 20
show_correlations, 22
show_weights, 23
∗ **spca**
aggregate_by_group, 4
change_sign, 5
compare_spca, 6
is.spca, 10
new_spca, 13
plot.pca, 16
print.pca, 19
show_correlations, 22
show_weights, 23
spca, 25
spca_object, 27
summary.pca, 29

aggregate_by_group, 4, *6*, *8*, *10, 11*, *13, 14*, *16*, *18*, *20, 21*, *23, 24*, *27, 28*, *30*
aggregate_by_group(), *3*

change_loadings_sign_spca, *5, 6*, *8*, *11*, *14*, *18*, *20*, *23, 24*, *27, 28*, *30*
change_loadings_sign_spca(), *3*
change_sign, 5, *5*, *8*, *10, 11*, *13, 14*, *16*, *18*, *20, 21*, *23, 24*, *27, 28*, *30*
change_sign(), *3*
change_weights_sign_spca, *5, 6*, *8*, *11*, *14*, *18*, *20*, *23, 24*, *27, 28*, *30*
change_weights_sign_spca(), *3*

compare_spca, *5*, 6, *6*, *11*, *14*, *18*, *20*, *23, 24*, *27, 28*, *30*
compare_spca(), *3*

holzinger, 8
holzinger_scales, 9

is.pca, *5, 6*, 10, *13*, *16*, *20, 21*, *23, 24*
is.pca(), *3*
is.spca, *5, 6*, *8*, 10, *14*, *18*, *20*, *23, 24*, *27, 28*, *30*
is.spca(), *3*

mp_qqplot, *5, 6*, *10*, 11, *15, 16*, *20, 21*, *23, 24*
mp_qqplot(), *3*

new_spca, *5, 6*, *8*, *11*, 13, *18*, *20*, *23, 24*, *27, 28*, *30*
new_spca(), *3*, *27, 28*

pca, *5, 6*, *10*, *13*, 14, *20, 21*, *23, 24*
pca(), *3*, *12, 13*, *21*, *27, 28*
plot(), *3*
plot.pca, *5, 6*, *8*, *11*, *14*, 16, *20*, *23, 24*, *27, 28*, *30*
plot.spca *(*plot.pca*)*, 16
print(), *3*
print.pca, *5, 6*, *8*, *10, 11*, *13, 14*, *16*, *18*, 19, *21*, *23, 24*, *27, 28*, *30*
print.spca *(*print.pca*)*, 19

scree_plot, *5, 6*, *10*, *13*, *16*, 20, *20*, *23, 24*
scree_plot(), *3*
show_contributions_spca, *5, 6*, *8*, *11*, *14*, *18*, *20*, *23, 24*, *27, 28*, *30*
show_correlations, *5, 6*, *8*, *10, 11*, *13, 14*, *16*, *18*, *20, 21*, 22, *24*, *27, 28*, *30*
show_weights, *5, 6*, *8*, *10, 11*, *13, 14*, *16*, *18*, *20, 21*, 23, *23*, *27, 28*, *30*
show_weights(), *3*
spca, *5, 6*, *8*, *11*, *14*, *18*, *20*, *23, 24*, 25, *28*, *30*

spca(), *3*, *13*, *21*, *27, 28*
spca-package, 2
spca_object, *3*, *5, 6*, *8*, *11*, *14*, *16*, *18*, *20*, *23, 24*, 27, *27*, *30*
spca_screeplot, *5, 6*, *10*, *13*, *16*, *20, 21*, *23, 24*
spca_screeplot(), *3*
summary(), *3*
summary.pca, *5, 6*, *8*, *11*, *14*, *18*, *20*, *23, 24*, *27, 28*, 29
summary.spca, *6*
summary.spca *(*summary.pca*)*, 29

wachter_qqplot, *5, 6*, *10*, *13*, *16*, *20, 21*, *23, 24*
wachter_qqplot(), *3*