Package: spca (via r-universe)

July 11, 2026

Type Package

Title Least Squares Sparse Principal Components Analysis

Version 1.1.1

Date 2026-06-03

Description Implements least-squares sparse principal component analysis (LS-SPCA). 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, bench, R.rsp

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 2026-07-06 00:16:19 UTC

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

RemoteRef HEAD

RemoteSha 5f0f2aa47dcb2f4024ff41dfb74de870ebdd2ff6

Contents

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

Description

The package provides functions to compute LS-SPCA solutions, where sparsity is imposed to Pearson's PCA's least-squares reconstruction objective.

Details

LS-SPCA is different for other SPCA methods that compute sparse PCs with maximal variance. Details about LS-SPCA can be found in the articles cited below and in the extended vignette.

This release accompanies the related article and is intended to support full reproduction of the results reported therein.

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

Fitting functions

- `spca()` Computes LS-SPCA solutions from a data or covariance/correlation matrix. Returns an spca_object of class `spca`.
- `pca()` Computes PCA solutions from a data or covariance/correlation matrix. Returns an spca_object of class `spca`.

S3 methods for objects of class `spca` include: `pca()` returns PCA results as an `spca` object. **methods**

- print()
- plot()
- summary()

Utilities

- is.spca() Verifies if an object inherits from class spca.
- compare_spca() Compares two or more LS-SPCA solutions numerically and visually.
- new_spca() Creates an spca object from a set of loadings.
- aggregate_by_group() Sums loadings or contributions wrt an index vector.
- show_contributions_spca() Prints the nonzero contributions separately for each sPC.
- change_loadings_sign_spca() Changes the sign of the loadings and all related elements in an 'spca object.
- spca_screeplot() and wachter_qqplot() Diagnostic plots usefull to determine the number of components to retain in PCA.

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 loadings. *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 loadings or contributions by group*

Description

Compute group-level sums from a vector, matrix, data frame, or spca object according to a grouping variable.

Usage

```
aggregate_by_group(
  X,
  groups,
  only_nonzero = TRUE,
  contributions = TRUE,
  digits = ifelse(contributions, 1, 3),
  print_table = TRUE,
  return_table = FALSE
)
```

Arguments

X	An spca object, numeric vector, numeric matrix, or numeric data frame containing loadings or contributions.
groups	A vector or factor with one group label per variable. Its length must equal length(X) when X is a vector, or nrow(X) when X is a matrix, data frame, or spca object.
only_nonzero	A logical value (default TRUE). If TRUE, groups with zero total absolute contribution or loading are removed from the output.
contributions	A logical value (default TRUE). If TRUE, aggregate and print values as percentage contributions when possible. If X is an spca object, X$contributions is used when contributions = TRUE; otherwise, X$loadings is used.
digits	An integer scalar (default 1 when contributions = TRUE, otherwise 3). Number of digits used for rounding in the printed output.
print_table	A logical value (default TRUE). If TRUE, print the aggregated table.
return_table	A logical value (default FALSE). If TRUE, return the aggregated table.

Details

If contributions = TRUE but the input values do not sum to one in absolute value, contributions is set to FALSE and digits is set to 3. Aggregated sums are not expected to retain the unit-norm property of the original loadings or contributions.

Value

Invisibly returns the aggregated numeric vector or matrix by default. If return_table = TRUE, returns the same object visibly. Rows correspond to groups and columns correspond to components when X is a matrix, data frame, or spca object.

Examples

```
data(holzinger)
data(holzinger_scales)
ho_cspca = spca(holzinger, n_comps = 2)
aggregate_by_group(ho_cspca, groups = holzinger_scales)
```

change_loadings_sign_spca *Change the sign of selected loadings in an spca object*

Description

Flip the sign of selected sparse principal components in an spca object.

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 flipped.

Details

The function multiplies by -1 the selected columns of loadings and contributions. It also updates loadings_list, scores, and the corresponding rows and columns of spc_cor when these elements are present. This is useful because principal components and sparse principal components are defined only up to sign.

Value

The modified spca_obj, with the selected components sign-flipped.

See Also

Other spca: compare_spca(), is.spca(), new_spca(), plot.spca(), print.spca(), show_contributions_spca(), spca(), spca_object, summary.spca()

compare_spca *Compare two or more spca solutions*

Description

Plot loadings 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_loadings = TRUE,
  plot_type = c("bars", "points"),
  methods_names = NULL,
  x_axis_var_names = TRUE,
  col_grouplines = "red",
  color_scale = c("ggplot", "cbb", "printsafe", "bw"),
  col_short_names = TRUE,
  print_tables = TRUE,
  print_loadings = 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 loadings.
only_nonzero	A logical value (default TRUE). If TRUE, only variables with at least one nonzero loading 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 loadings plot.
plot_loadings	A logical value (default TRUE). If TRUE, plot the loadings 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 TRUE). If TRUE, show variable names on the x axis of the loadings 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_loadings.

print_loadings
: A logical value (default TRUE). If TRUE, print the loadings or contributions table.

show_plot
: A logical value (default TRUE). If TRUE, show the loadings or contributions plot.

return_tables
: A logical value (default FALSE). If TRUE, return the loadings or contributions matrix and the raw summary matrix.

return_plot
: A logical value (default FALSE). If TRUE, return the loadings 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 loadings or contributions plot.

See Also

Other spca: change_loadings_sign_spca(), is.spca(), new_spca(), plot.spca(), print.spca(), show_contributions_spca(), 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` *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.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 loadings, contributions, explained-variance summaries, component counts, cardinalities, loading 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: `change_loadings_sign_spca()`, `compare_spca()`, `new_spca()`, `plot.spca()`, `print.spca()`, `show_contributions_spca()`, `spca()`, `spca_object`, `summary.spca()`

Examples

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

new_spca *Construct an spca object from a set of loadings*

Description

Build an object of class spca from a loadings 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 loadings.
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 loadings.

Value

An spca object.

See Also

Other spca: change_loadings_sign_spca(), compare_spca(), is.spca(), plot.spca(), print.spca(), show_contributions_spca(), 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 *Compute 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

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 FALSE). If TRUE, produce a scree plot.
qq_plot	A logical value (default TRUE). If TRUE, produce a Wachter QQ plot with wachter_qqplot.
nrow_data	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). 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 loadings.

Value

An spca_object with an additional eigenvalues vector containing the eigenvalues up to the rank used by the selected backend.

See Also

Other pca: 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 loadings, 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_loadings = NULL,
  variable_groups = NULL,
  plot_title = NULL,
  return_plot = FALSE,
  show_plot = TRUE,
 controls = list(color_scale = c("ggplot", "cbb", "printsafe", "bw"), variable_names =
  NULL, 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.
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 loadings.
only_nonzero	A logical value (default TRUE). If TRUE, plot only variables with at least one nonzero loading.
pc_loadings	A numeric matrix, data frame, or NULL (default NULL). Optional PCA loadings 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.
...	Further arguments. These are currently unused and trigger an error if supplied.

Details

If pc_loadings 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 NULL). If NULL, row names of the loading 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: change_loadings_sign_spca(), compare_spca(), is.spca(), new_spca(), print.spca(), show_contributions_spca(), 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` *Print an spca object*

Description

Print sparse loadings, 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

x	An object of class spca.
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 loading or contribution whose absolute value is greater than or equal to thresh_card.
contributions	A logical value (default TRUE). If TRUE, print loadings scaled to unit L_1 norm as percentage contributions; otherwise, print L2 unit loadings.

digits	An integer scalar (default 3). Number of decimal places used when printing loadings. 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.
...	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: change_loadings_sign_spca(), compare_spca(), is.spca(), new_spca(), plot.spca(), show_contributions_spca(), spca(), spca_object, summary.spca()

Examples

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

---

show_contributions_spca

Show nonzero contributions by component

Description

Convert the nonzero loadings stored in an spca object into percentage contributions for selected components.

Usage

```
show_contributions_spca(
  spca_obj,
  cols = NULL,
  print_list = TRUE,
  return_list = FALSE
)
```

Arguments

spca_obj	An object of class spca.
cols	An integer vector or NULL (default NULL). Components to show. If NULL, all components are shown. If a single component is supplied, a single contribution vector is produced.
print_list	A logical value (default TRUE). If TRUE, the contribution list or vector is printed.
return_list	A logical value (default FALSE). If TRUE, the contribution list or vector is returned.

Value

If return_list = TRUE, returns the selected contributions. Otherwise, returns NULL invisibly.

See Also

Other spca: change_loadings_sign_spca(), compare_spca(), is.spca(), new_spca(), plot.spca(), print.spca(), spca(), spca_object, summary.spca()

Examples

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

---

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_loading = FALSE,
  eps_pm_loading = 1e-04,
  maxiter_pm_loading = 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 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.
pm_loading	A logical value (default FALSE). If TRUE, use the power method for PC and sparse-loading eigenvectors.

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

`maxiter_pm_loading` A positive integer scalar (default `1000`). Maximum number of iterations for `pm_loading`.

`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.

The returned object is documented in spca_object.

Value

An object of class `spca`.

See Also

Other spca: `change_loadings_sign_spca()`, `compare_spca()`, `is.spca()`, `new_spca()`, `plot.spca()`, `print.spca()`, `show_contributions_spca()`, `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 by new_spca()..

Components

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

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

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

n_comps Number of sPCs.

cardinality Number of nonzero loadings 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.

loadings_list List of nonzero loading vectors, one per sPC.

spc_cor $n_comps \times n_comps$ correlation matrix of the sPC scores.

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

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

parameters List of parameters used to compute the fit.

call Matched call used to compute the fit.

See Also

Other spca: change_loadings_sign_spca(), compare_spca(), is.spca(), new_spca(), plot.spca(), print.spca(), show_contributions_spca(), spca(), summary.spca()

spca_screeplot *Plot eigenvalues in a scree plot*

Description

Plot the first nplot eigenvalues against component order.

Usage

```
spca_screeplot(
  eigenvalues,
  nplot = NULL,
  ylab = "eigenvalues",
  addtitle = TRUE,
  show_plot = TRUE,
  return_plot = FALSE
)
```

Arguments

eigenvalues	A numeric vector of eigenvalues.
nplot	An integer scalar or NULL (default NULL). Number of leading eigenvalues to plot. If NULL, all eigenvalues are plotted.
ylab	A character scalar (default "eigenvalues"). Label for the y axis.
addtitle	A logical value (default TRUE). If TRUE, add a plot title.
show_plot	A logical value (default TRUE). If TRUE, print the plot.
return_plot	A logical value (default FALSE). If TRUE, return the ggplot object.

Value

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

See Also

Other pca: pca(), wachter_qqplot()

Examples

```
data(holzinger)
ho_pca = pca(holzinger,  qq_plot = FALSE)
 spca_screeplot(ho_pca$eigenvalues)
```

summary.spca *Summarize an spca object*

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_load = FALSE,
  cor_with_pc = FALSE,
  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 loadings.
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_load	A logical value (default FALSE). If TRUE, include the minimum nonzero loading 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 loadings.
Min load/Min cont	The minimum absolute value of the nonzero loadings 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: `change_loadings_sign_spca()`, `compare_spca()`, `is.spca()`, `new_spca()`, `plot.spca()`, `print.spca()`, `show_contributions_spca()`, `spca()`, `spca_object`

Examples

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

---

wachter_qqplot *Wachter QQ plot for eigenvalues*

Description

Produce a QQ plot comparing observed eigenvalues with Marchenko–Pastur (Wachter) theoretical quantiles.

Usage

```
wachter_qqplot(
  eigenvalues,
  p = NULL,
  n,
  gamma,
  cor = TRUE,
  common_var = 1,
```

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

Arguments

eigenvalues	A numeric vector of eigenvalues, assumed to be sorted in decreasing order.
p	An integer scalar or NULL (default NULL). Number of variables. If NULL, length(eigenvalues) is used.
n	An integer scalar. Sample size.
gamma	A numeric scalar. Aspect ratio. If missing, n / p is used.
cor	A logical value (default TRUE). Currently accepted for compatibility; the plotted quantiles are controlled by common_var.
common_var	A positive numeric scalar (default 1). Common variance of the variables. Use this when the variables were rescaled to unit variance. See Details.
nplot	An integer scalar or NULL (default NULL). Number of leading eigenvalues to include. If NULL, all eigenvalues are included.
n_fitline	An integer scalar or NULL (default NULL). If positive, fit a least-squares line using the last n_fitline points. If negative, exclude the largest abs(n_fitline) values from the fitted line.
addtitle	A logical value (default TRUE). If TRUE, add a plot title.
show_plot	A logical value (default TRUE). If TRUE, print the plot.
return_plot	A logical value (default FALSE). If TRUE, return the ggplot object.

Details

The QQ plot is based on the Marchenko–Pastur distribution of the eigenvalues of a random covariance matrix generated from variables with a common variance. If the data set or covariance matrix comes from variables with different variances, the QQ plot is not valid. A simple introduction to the QQ plot can be found at https://brainder.org/tag/wachter-test/; see the extended vignette for references.

Value

If return_plot = TRUE, returns a ggplot object. Otherwise, returns NULL invisibly.

See Also

Other pca: pca(), spca_screeplot()

Examples

```
data(holzinger)
ho_pca = pca(holzinger,  qq_plot = FALSE)
 wachter_qqplot(ho_pca$eigenvalues, p = ncol(holzinger), n = nrow(holzinger),
  cor = TRUE, n_fitline = -3)
```

Index