Package ‘ggbiplot’

January 9, 2024

Type Package

Title A Grammar of Graphics Implementation of Biplots

Version 0.6.2

Date 2024-01-06

Description A 'ggplot2' based implementation of biplots, giving a representation of a dataset in a two dimensional space accounting for the greatest variance, together with variable vectors showing how the data variables relate to this space. It provides a
replacement for stats::biplot(), but with many enhancements to control the analysis and graphical display. It implements
biplot and scree plot methods which can be used with the results of prcomp(), princomp(), FactoMineR::PCA(), ade4::dudi.pca() or MASS::lda() and can be customized using 'ggplot2' techniques.

Depends R (>= 3.5.0), ggplot2

Imports scales

Suggests corrplot, dplyr, MASS, broom, tidyr

License GPL-2

Encoding UTF-8

Language en-US

URL https://github.com/friendly/ggbiplot,
https://friendly.github.io/ggbiplot/

BugReports https://github.com/friendly/ggbiplot/issues

RoxygenNote 7.2.3

LazyData true

NeedsCompilation no

Author Vincent Q. Vu [aut] (<https://orcid.org/0000-0002-4689-0497>),
Michael Friendly [aut, cre] (<https://orcid.org/0000-0002-3237-0941>),
Aghasi Tavadyan [ctb]

Maintainer Michael Friendly <friendly@yorku.ca>

Repository CRAN

Date/Publication 2024-01-08 23:10:11 UTC

R topics documented:

`crime` *U. S. Crimes*

Description

This dataset gives rates of occurrence (per 100,000 people) various serious crimes in each of the 50 U. S. states, originally from the United States Statistical Abstracts (1970). The data were analyzed by John Hartigan (1975) in his book *Clustering Algorithms* and were later reanalyzed by Friendly (1991).

Usage

```
data(crime)
```

Format

A data frame with 50 observations on the following 10 variables.

`state` state name, a character vector

`murder` a numeric vector

`rape` a numeric vector

`robbery` a numeric vector

`assault` a numeric vector

`burglary` a numeric vector

`larceny` a numeric vector

`auto` auto thefts, a numeric vector

`st` state abbreviation, a character vector

`region` region of the U.S., a factor with levels `Northeast` `South` `North Central` `West`

Source

The data are originally from the United States Statistical Abstracts (1970). This dataset also appears in the SAS/Stat Sample library, *Getting Started Example for PROC PRINCOMP*, `https://support.sas.com/documentation/onlinedoc/stat/ex_code/131/princgs.html`, from which the current copy was derived.

References

Friendly, M. (1991). *SAS System for Statistical Graphics*. SAS Institute.

Hartigan, J. A. (1975). *Clustering Algorithms*. John Wiley and Sons.

Examples

```
data(crime)
library(ggplot2)
crime.pca <-
  crime |>
  dplyr::select(where(is.numeric)) |>
  prcomp(scale. = TRUE)

ggbiplot(crime.pca,
     labels = crime$st ,
     circle = TRUE,
     varname.size = 4,
     varname.color = "red") +
 theme_minimal(base_size = 14)
```

---

get_SVD *Extract the SVD components from a PCA-like object*

Description

Biplots are based on the Singular Value Decomposition, which for a data matrix is

$$\mathbf{X}/\sqrt{n} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

but these are computed and returned in quite different forms by various PCA-like methods. This function provides a common interface, returning the components with standard names.

Usage

```
get_SVD(pcobj)
```

Arguments

pcobj	an object returned by prcomp, princomp, PCA, dudi.pca, or lda

Value

A list of four elements

n The sample size on which the analysis was based

U Left singular vectors, giving observation scores

D vector of singular values, the diagonal elements of the matrix $\mathbf{D}$, which are also the square roots of the eigenvalues of $\mathbf{X}\mathbf{X}'$

V Right singular vectors, giving variable loadings

Examples

```
data(crime)
crime.pca <-
  crime |>
  dplyr::select(where(is.numeric)) |>
  prcomp(scale. = TRUE)

crime.svd <- get_SVD(crime.pca)
names(crime.svd)
crime.svd$D
```

`ggbiplot` *Biplot for Principal Components using ggplot2*

Description

A biplot simultaneously displays information on the observations (as points) and the variables (as vectors) in a multidimensional dataset. The 2D biplot is typically based on the first two principal components of a dataset, giving a rank 2 approximation to the data. The “bi” in biplot refers to the fact that two sets of points (i.e., the rows and columns of the data matrix) are visualized by scalar products, not the fact that the display is usually two-dimensional.

The biplot method for principal component analysis was originally defined by Gabriel (1971, 1981). Gower & Hand (1996) give a more complete treatment. Greenacre (2010) is a practical user-oriented guide to biplots. Gower et al. (2011) is the most up to date exposition of biplot methodology.

This implementation handles the results of a principal components analysis using `prcomp`, `princomp`, `PCA` and `dudi.pca`; also handles a discriminant analysis using `lda`.

Usage

```
ggbiplot(
  pcobj,
  choices = 1:2,
  scale = 1,
  pc.biplot = TRUE,
  obs.scale = 1 - scale,
  var.scale = scale,
  var.factor = 1,
  groups = NULL,
  point.size = 1.5,
  ellipse = FALSE,
  ellipse.prob = 0.68,
  ellipse.linewidth = 1.3,
  ellipse.fill = TRUE,
  ellipse.alpha = 0.25,
  labels = NULL,
  labels.size = 3,
  alpha = 1,
  var.axes = TRUE,
  circle = FALSE,
  circle.prob = 0.68,
  varname.size = 3,
  varname.adjust = 1.25,
  varname.color = "black",
  varname.abbrev = FALSE,
  axis.title = "PC",
  ...
)
```

Arguments

`pcobj` an object returned by `prcomp`, `princomp`, `PCA`, `dudi.pca`, or `lda`

`choices` Which components to plot? An integer vector of length 2.

`scale` Covariance biplot (`scale = 1`), form biplot (`scale = 0`). When `scale = 1` (the default), the inner product between the variables approximates the covariance and the distance between the points approximates the Mahalanobis distance.

`pc.biplot` Logical, for compatibility with `biplot.princomp()`. If `TRUE`, use what Gabriel (1971) refers to as a "principal component biplot", with $\alpha = 1$ and observations scaled up by $sqrt(n)$ and variables scaled down by $sqrt(n)$. Then inner products between variables approximate covariances and distances between observations approximate Mahalanobis distance.

`obs.scale` Scale factor to apply to observations

`var.scale` Scale factor to apply to variables

`var.factor` Factor to be applied to variable vectors after scaling. This allows the variable vectors to be reflected (`var.factor = -1`) or expanded in length (`var.factor > 1`) for greater visibility. `reflect` provides a simpler way to reflect the variables.

`groups` Optional factor variable indicating the groups that the observations belong to. If provided the points will be colored according to groups and this allows data ellipses also to be drawn when `ellipse = TRUE`.

`point.size` Size of observation points.

`ellipse` Logical; draw a normal data ellipse for each group?

`ellipse.prob` Coverage size of the data ellipse in Normal probability

`ellipse.linewidth`
Thickness of the line outlining the ellipses

`ellipse.fill` Logical; should the ellipses be filled?

`ellipse.alpha` Transparency value (0 - 1) for filled ellipses

`labels` Optional vector of labels for the observations. Often, this will be specified as the `row.names()` of the dataset.

`labels.size` Size of the text used for the point labels

`alpha` Alpha transparency value for the points (0 = transparent, 1 = opaque) `var.axes` logical; draw arrows for the variables?

`circle` draw a correlation circle? (only applies when prcomp was called with `scale = TRUE` and when `var.scale = 1`)

`circle.prob` Size of the correlation circle

`varname.size` Size of the text for variable names

`varname.adjust` Adjustment factor the placement of the variable names, >= 1 means farther from the arrow

`varname.color` Color for the variable vectors and names

`varname.abbrev` logical; whether or not to abbreviate the variable names, using `abbreviate`.

`axis.title` character; the prefix used as the axis labels. Default: `"PC"`.

`...` other arguments passed down

Details

The biplot is constructed by using the singular value decomposition (SVD) to obtain a low-rank approximation to the data matrix $\mathbf{X}_{n \times p}$ (centered, and optionally scaled to unit variances) whose n rows are the observations and whose p columns are the variables.

Using the SVD, the matrix $\mathbf{X}$, of rank $r \leq p$ can be expressed *exactly* as

$$\mathbf{X} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}' = \Sigma_i^r \lambda_i \mathbf{u}_i \mathbf{v}_i' \ ,$$

where

- $\mathbf{U}$ is an $n \times r$ orthonormal matrix of observation scores; these are also the eigenvectors of $\mathbf{X}\mathbf{X}'$,
- $\mathbf{\Lambda}$ is an $r \times r$ diagonal matrix of singular values, $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_r$
- $\mathbf{V}$ is an $r \times p$ orthonormal matrix of variable weights and also the eigenvectors of $\mathbf{X}'\mathbf{X}$.

Then, a rank 2 (or 3) PCA approximation $\widehat{\mathbf{X}}$ to the data matrix used in the biplot can be obtained from the first 2 (or 3) singular values λ_i and the corresponding $\mathbf{u}_i, \mathbf{v}_i$ as

$$\mathbf{X} \approx \widehat{\mathbf{X}} = \lambda_1 \mathbf{u}_1 \mathbf{v}_1' + \lambda_2 \mathbf{u}_2 \mathbf{v}_2' \ .$$

The variance of $\mathbf{X}$ accounted for by each term is λ_i^2.

The biplot is then obtained by overlaying two scatterplots that share a common set of axes and have a between-set scalar product interpretation. Typically, the observations (rows of $\mathbf{X}$) are represented as points and the variables (columns of $\mathbf{X}$) are represented as vectors from the origin.

The `scale` factor, α allows the variances of the components to be apportioned between the row points and column vectors, with different interpretations, by representing the approximation $\widehat{\mathbf{X}}$ as the product of two matrices,

$$\widehat{\mathbf{X}} = (\mathbf{U}\mathbf{\Lambda}^{\alpha})(\mathbf{\Lambda}^{1-\alpha}\mathbf{V}')$$

The choice $\alpha = 1$, assigning the singular values totally to the left factor, gives a distance interpretation to the row display and $\alpha = 0$ gives a distance interpretation to the column display. $\alpha = 1/2$ gives a symmetrically scaled biplot.

When the singular values are assigned totally to the left or to the right factor, the resultant coordinates are called *principal coordinates* and the sum of squared coordinates on each dimension equal the corresponding singular value. The other matrix, to which no part of the singular values is assigned, contains the so-called *standard coordinates* and have sum of squared values equal to 1.0.

Value

a ggplot2 plot object of class `c("gg", "ggplot")`

Author(s)

Vincent Q. Vu.

References

Gabriel, K. R. (1971). The biplot graphical display of matrices with application to principal component analysis. *Biometrika*, **58**, 453–467. doi:10.2307/2334381.

Gabriel, K. R. (1981). Biplot display of multivariate matrices for inspection of data and diagnosis. In V. Barnett (Ed.), *Interpreting Multivariate Data*. London: Wiley.

Greenacre, M. (2010). *Biplots in Practice*. BBVA Foundation, Bilbao, Spain. Available for free at `https://www.fbbva.es/microsite/multivariate-statistics/`.

J.C. Gower and D. J. Hand (1996). *Biplots*. Chapman & Hall.

Gower, J. C., Lubbe, S. G., & Roux, N. J. L. (2011). *Understanding Biplots*. Wiley.

See Also

`reflect`, `ggscreeplot`; `biplot` for the original stats package version; `fviz_pca_biplot` for the factoextra package version.

Examples

```
data(wine)
library(ggplot2)
wine.pca <- prcomp(wine, scale. = TRUE)
ggbiplot(wine.pca,
         obs.scale = 1, var.scale = 1,
         varname.size = 4,
         groups = wine.class,
         ellipse = TRUE, circle = TRUE)

data(iris)
iris.pca <- prcomp (~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
                    data=iris,
                    scale. = TRUE)
ggbiplot(iris.pca, obs.scale = 1, var.scale = 1,
         groups = iris$Species, point.size=2,
         varname.size = 5,
         varname.color = "black",
         varname.adjust = 1.2,
         ellipse = TRUE,
        circle = TRUE) +
  labs(fill = "Species", color = "Species") +
  theme_minimal(base_size = 14) +
  theme(legend.direction = 'horizontal', legend.position = 'top')
```

ggscreeplot *Screeplot for Principal Components*

Description

Produces scree plots (Cattell, 1966) of the variance proportions explained by each dimension against dimension number from various PCA-like dimension reduction techniques.

Usage

```
ggscreeplot(
  pcobj,
  type = c("pev", "cev"),
  size = 4,
  shape = 19,
  color = "black",
  linetype = 1,
  linewidth = 1
)
```

Arguments

`pcobj`	an object returned by `prcomp`, `princomp`, `PCA`, `dudi.pca`, or `lda`
`type`	the type of scree plot, one of `c('pev', 'cev')`. `'pev'` plots the proportion of explained variance, i.e. the eigenvalues divided by the trace. `'cev'` plots the cumulative proportion of explained variance, i.e. the partial sum of the first k eigenvalues divided by the trace.
`size`	point size
`shape`	shape of the points. Default: 19, a filled circle.
`color`	color for points and line. Default: `"black"`.
`linetype`	type of line
`linewidth`	width of line

Value

A ggplot2 object with the aesthetics `x = PC, y = yvar`

References

Cattell, R. B. (1966). The Scree Test For The Number Of Factors. *Multivariate Behavioral Research*, 1, 245–276.

Examples

```
data(wine)
wine.pca <- prcomp(wine, scale. = TRUE)
ggscreeplot(wine.pca)

# show horizontal lines for 80, 90% of cumulative variance
ggscreeplot(wine.pca, type = "cev") +
  geom_hline(yintercept = c(0.8, 0.9), color = "blue")

# Make a fancy screeplot, higlighting the scree starting at component 4
data(crime)
crime.pca <-
  crime |>
  dplyr::select(where(is.numeric)) |>
  prcomp(scale. = TRUE)

(crime.eig <- crime.pca |>
   broom::tidy(matrix = "eigenvalues"))

ggscreeplot(crime.pca) +
  stat_smooth(data = crime.eig |> dplyr::filter(PC>=4),
              aes(x=PC, y=percent), method = "lm",
              se = FALSE,
              fullrange = TRUE)
```

reflect *Reflect Columns in a Principal Component-like Object*

Description

Principle component-like objects have variable loadings (the eigenvectors of the covariance/correlation matrix) whose signs are arbitrary, in the sense that a given column can be reflected (multiplied by -1) without changing the fit.

Usage

```
reflect(pcobj, columns = 1:2)
```

Arguments

pcobj	an object returned by prcomp, princomp, PCA, or lda
columns	a vector of indices of the columns to reflect

Details

This function allows one to reflect any columns of the variable loadings (and corresponding observation scores). Coordinates for quantitative supplementary variables are also reflected if present. This is often useful for interpreting a biplot, for example when a component (often the first) has all negative signs.

Value

The pca-like object with specified columns of the variable loadings and observation scores multiplied by -1.

Author(s)

Michael Friendly

See Also

`prcomp`, `princomp`, `PCA`, `lda`

Examples

```
data(crime)
crime.pca <-
  crime |>
  dplyr::select(where(is.numeric)) |>
  prcomp(scale. = TRUE)

 biplot(crime.pca)

 crime.pca <- reflect(crime.pca)  # reflect columns 1:2
 biplot(crime.pca)
```

`wine` *Wine dataset*

Description

Results of a chemical analysis of wines grown in the same region in Italy, derived from three different cultivars. The analysis determined the quantities of 13 chemical constituents found in each of the three types of wines.

The grape varieties (cultivars), ’barolo’, ’barbera’, and ’grignolino’, are indicated in `wine.class`.

Usage

```
data(wine)
```

Format

A `wine` data frame consisting of 178 observations (rows) and 13 columns and vector `wine.class` of factors indicating the cultivars. The variables are:

`Alcohol` a numeric vector

`MalicAcid` Malic acid, a numeric vector

`Ash` Ash, a numeric vector

`AlcAsh` Alcalinity of ash, a numeric vector

`Mg` Magnesium, a numeric vector

`Phenols` total phenols, a numeric vector

`Flav` Flavanoids, a numeric vector

`NonFlavPhenols` Nonflavanoid phenols, a numeric vector

`Proa` Proanthocyanins, a numeric vector

`Color` Color intensity, a numeric vector

`Hue` a numeric vector

`OD` D280/OD315 of diluted wines, a numeric vector

`Proline` a numeric vector

Source

UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/datasets/Wine)

Examples

```
data(wine)
table(wine.class)

wine.pca <- prcomp(wine, scale. = TRUE)
ggscreeplot(wine.pca)
ggbiplot(wine.pca,
         obs.scale = 1, var.scale = 1,
         groups = wine.class, ellipse = TRUE, circle = TRUE)
```

Index