

Package ‘bipl5’

May 7, 2026

Title Construct Reactive Calibrated Axes Biplots

Version 1.0.2

Description A modern view on the principal component analysis biplot with calibrated axes. Create principal component analysis biplots rendered in HTML with significant reactivity embedded within the plot. Furthermore, the traditional biplot view is enhanced by translated axes with inter-class kernel densities superimposed. For more information on biplots, see Gower, J.C., Lubbe, S. and le Roux, N.J. (2011, ISBN: 978-0-470-01255-0).

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

Imports cluster, crayon (>= 1.5.2), htmlwidgets (>= 1.6.2), knitr (>= 1.43), methods, plotly (>= 4.10.2)

Depends R (>= 2.10)

NeedsCompilation no

Author Ruan Buys [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0001-8527-8631>>),
Carel van der Merwe [aut, ths] (ORCID:
<<https://orcid.org/0000-0003-0676-8240>>)

Maintainer Ruan Buys <ruanbuys8@gmail.com>

Repository CRAN

Date/Publication 2023-11-02 15:50:05 UTC

Contents

bipl5	2
colorpal	3
FMbiplot	3
PCAbiplot	4
plot.bipl5	6
predict.bipl5	6
print.bipl5	7

Symbol_List	8
TDAbiplot	8

Index	12
--------------	-----------

bipl5	<i>bipl5: Constructing Reactive Calibrated Axes Biplots</i>
-------	---

Description

A modern view on PCA biplot with calibrated axes. Create PCA biplots rendered in HTML with significant reactivity embedded on the plot. Furthermore, the traditional biplot view is enhanced by translated axes with interclass kernel densities superimposed.

Details

Package:	bipl5
Type:	Package
Version:	0.0.0.9400
Date:	16-10-2023
License:	MIT
LazyLoad:	TRUE

Author(s)

- Ruan Buys (Maintainer)
- Carel van der Merwe

Core Functions

- [PCAbiplot](#)
- [TDAbiplot](#)
- [FMbiplot](#)
- [predict.bipl5](#)
- [print.bipl5](#)

Code Availability

The newest version of the package can be obtained on GitHub: <https://github.com/RuanBuys/bipl5>

colorpal	<i>bpl5 default color scales</i>
----------	----------------------------------

Description

bpl5 default color scales

Usage

```
colorpal(number = 16)
```

Arguments

number Number of distinct colors to return

Value

A vector of colors

Examples

```
colorpal()
```

FMbiplot	<i>Determine various measures of fit for the PCA biplot</i>
----------	---

Description

Print various measures of fit of the biplot display to the console

Usage

```
FMbiplot(x)
```

Arguments

x An object of class bpl5

Value

A list returned invisibly containing the following fit measures:

- Cumulative Predictivity
- Marginal Predictivity for the first three principal components
- Marginal Adequacy for the first three principal components
- Overall quality of display

Examples

```
x<-PCAbiplot(iris[,-5])
FMbiplot(x)
```

PCAbiplot

Construct a rank-2 PCA biplot

Description

Rank-2 PCA biplots are constructed based on a combination of the first three principal components.

Usage

```
PCAbiplot(
  x,
  group = NULL,
  scale = TRUE,
  basis = 1:2,
  symbol = "circle",
  color = NULL,
  build_plot = TRUE
)
```

Arguments

<code>x</code>	A numeric matrix or data frame of size $n \times p$
<code>group</code>	Vector of size n representing the class of each observation in x
<code>scale</code>	Boolean: indicating whether the data matrix should be standardized before SVD is performed. Similar to the <code>cor</code> argument in princomp
<code>basis</code>	A vector specifying which eigenvectors serve as basis for the plot. Currently only a biplot of rank 2 is supported
<code>symbol</code>	Plotting symbol to be used per class group
<code>color</code>	Colors to be utilized per class group
<code>build_plot</code>	Boolean, indicating whether the biplot should be drawn or not. Mostly used in internal function calls serve as basis for the biplot. Currently only a biplot of rank 2 is supported

Details

The method performs Principal Component Analysis (PCA) on the input data and constructs both a traditional biplot using vector representation and with calibrated axes. The data is clustered together on the display by the `group` parameter. The `scale` parameter determines if SVD is performed on the covariance matrix or correlation of x . It is highly recommended to set `scale=TRUE` as the rendered display is sensitive to the scaling in x .

By default three sets of principal components are used for the scaffolding axes, namely: 1 and 2, 1 and 3, and 2 and 3. The function constructs these biplots in the [plot_ly](#) graphing library with reactivity embedded on the display. The following features are available on the display:

- A dropdown menu to change the principal components used to construct the display. Currently only the first three pairwise are supported.
- A button to give fit statistics of the biplot. Once clicked, a table is added to give the adequacy and predictivity of each axis for the display.
- A button that inserts a graph depicting the cumulative predictivity of each axis against the dimension of the biplot. See [FMbiplot](#) for the coordinates.
- A button that changes the display from vector representation of the variables, to calibrated axes in their original scale. The vector representation includes a unit circle around the origin.
- Prediction lines are inserted onto the display if an observation is clicked. The prediction lines can be removed by clicking on the legend entry.

Value

A named list of class `bip15` with the following attributes

<code>x</code>	A data frame which is the original input data
<code>Z</code>	A matrix of $n \times 2$ representing the coordinates of each observation on the biplot
<code>rank</code>	The rank of the approximated data
<code>scale</code>	Whether the data is standardized prior to performing dimension reduction
<code>group</code>	The grouping vector of the data
<code>mu</code>	The vector of column means of the input data
<code>stddev</code>	Vector of column standard deviations if the <code>scale</code> parameter is set to <code>TRUE</code> .
<code>PCA</code>	The singular value decomposition of the covariance/correlation matrix, see svd
<code>bip1</code>	The plotly graph displaying the biplot, see plot_ly
<code>Adequacy</code>	The adequacy of each axis displayed for each set of principal components
<code>Predictivity</code>	The predictivity of each axis displayed for each set of principal components

See Also

[print.bip15](#) to display the biplot graph and further see [TDAbiplot](#) to draw a biplot with calibrated density axes. Fit measures can be obtained by [FMbiplot](#)

Examples

```
## Construct a biplot with
x<-PCAbiplot(iris[,1:4],group=iris[,5])
#alternatively
print(x)

## Construct a biplot that preserves the correlation structure among the variables
y<-PCAbiplot(iris[, -5],group=iris[,5],scale=TRUE)
```

plot.bipl5	<i>Plot an object of class bipl5</i>
------------	--------------------------------------

Description

Plot an object of class bipl5

Usage

```
## S3 method for class 'bipl5'
plot(x, y = NULL, ...)
```

Arguments

x	An object of class bipl5
y	Unsupported
...	Unsupported

Value

A `plot_ly` graph containing the biplot

Examples

```
x<-PCAbiplot(iris[,-5])
plot(x)
```

predict.bipl5	<i>Method to obtain the predicted or fitted values of the biplot</i>
---------------	--

Description

Extract the fitted values of the biplot display

Usage

```
## S3 method for class 'bipl5'
predict(object, ...)
```

Arguments

object	An object of class bipl5 from which predicted values are to be obtained
...	Additional arguments to be passes to the kable function, contained in a list named kable.args

Value

The function invisibly returns the predicted values of the biplot display, and outputs the predicted values via the `kable` function

Examples

```
kable.args<-list()
kable.args$format<-"pipe"
x<-PCAbiplot(iris[,-5])
predict(x,kable.args)
```

`print.bipl5`

Default print method for an object of class bipl5

Description

Default print method for an object of class `bipl5`

Usage

```
## S3 method for class 'bipl5'
print(x, ...)
```

Arguments

<code>x</code>	Object of class <code>bipl5</code>
<code>...</code>	Additional parameters

Value

The object is returned invisibly

Examples

```
x<-PCAbiplot(iris[,1:4],group=iris[,5])
print.bipl5(x)
```

Symbol_List	<i>Retrieve all valid plotting symbols for plotly library</i>
-------------	---

Description

Retrieve all valid plotting symbols for plotly library

Usage

```
Symbol_List()
```

Value

A vector of all the valid plotting symbols used in the `plot_ly` library.

Examples

```
Symbol_List()
```

TDAbiplot	<i>Construct PCA biplots with translated calibrated density axes</i>
-----------	--

Description

Construct various rank-2 PCA biplots with translated axes based on a combination of the first three principal components.

Usage

```
TDAbiplot(
  x,
  dist = NULL,
  inflate = 1,
  alpha = 0.95,
  alpha_Elip = NULL,
  swop = FALSE,
  density.args = NULL,
  color = NULL,
  symbol = "circle"
)

## S3 method for class 'bip15'
TDAbiplot(
  x,
  dist = NULL,
  inflate = 1,
```

```

    alpha = 0.95,
    alpha_Elip = NULL,
    swop = FALSE,
    density.args = NULL,
    color = NULL,
    symbol = "circle"
)

```

Arguments

x	An object of class bip15. See PCAbiplot in this regard.
dist	Minimum distance between each axis. Default is roughly 12.5% of the plot diameter
inflate	Density inflation factor
alpha	Argument passes to alpha_Elip
alpha_Elip	A function taking two arguments, Z and alpha. The output of the function should be a two-column matrix of coordinates which will be used to construct an alpha-ellipse. See details below.
swop	Swop the direction which to which each axis is translated
density.args	Arguments to be passed to the density function
color	Colors to be utilized per class group
symbol	Plotting symbol to be used per class group

Details

This function produces a PCA biplot with translated calibrated axes. The function constructs this biplot in the [plot_ly](#) graphing library with reactivity embedded on the display. The following features are available on the display:

- A dropdown menu to change the principal components used to construct the display. Currently only the first three pairwise are supported.
- A button to give fit statistics of the biplot. Once clicked, a table is added to give the adequacy and predictivity of each axis for the display.
- A button that inserts a graph depicting the cumulative predictivity of each axis against the dimension of the biplot.
- Prediction lines are inserted onto the display if an observation is clicked. The prediction lines can be removed by clicking on the legend entry.

The `alpha_Elip` argument is used to subset the biplot plotting coordinates (Z) to remove the effect of outliers in the data. A common suggestion is to use an alphabag or on Convex hull peeling algorithm to strip away extreme points. The alpha-ellipse will be constructed around this data, and will impact the lengths of the calibrated axes.

Value

A named list of class `bip15`, see [PCAbiplot](#), with the following attributes:

<code>x</code>	A data frame which is the original input data
<code>Z</code>	A matrix of $n \times 2$ representing the coordinates of each observation on the biplot
<code>rank</code>	The rank of the approximated data
<code>scale</code>	Whether the data is standardized prior to performing dimension reduction
<code>group</code>	The grouping vector of the data
<code>mu</code>	The vector of column means of the input data
<code>stddev</code>	Vector of column standard deviations if the <code>scale</code> parameter is set to <code>TRUE</code> .
<code>PCA</code>	The singular value decomposition of the covariance/correlation matrix, see svd
<code>plot</code>	The plotly graph displaying the biplot, see plot_ly
<code>Adequacy</code>	The adequacy of each axis displayed for each set of principal components
<code>Predictivity</code>	The predictivity of each axis displayed for each set of principal components

See Also

[PCAbiplot](#) [FMbiplot](#)

Examples

```
## Simple illustration of a calibrated density axis biplot
x<-PCAbiplot(iris[,-5],group=iris[,5])
TDAbiplot(x,dist=1,inflate=1)

## Change the plotting characters of class-groups:
y<- x |> TDAbiplot(dist=1,inflate=1,symbol=c("circle","diamond","square"))

## Custom kernel densities can be drawn on the axes:
density.args<-list()
density.args$kernel <- "optcosine"
density.args$bw <- "sj"

y<- x |> TDAbiplot(dist=1,inflate=1,density.args=density.args)

## To lessen the effects of outliers, a smaller alpha-ellipse can be
## used to determine axis lengths. Define a function that strips away
## outliers, for example a convex hull peeling algorithm:

HullPeeling <- function(x,alpha) {
  n<-nrow(x)
  propinside<-1
  target<-1-alpha
  x2<-x
  while (propinside>target) {
    hull<-grDevices::chull(x2)
    x2old<-x2
    x2<-x2[-hull,]
  }
}
```

```
    propinside<-nrow(x2)/n
  }
  return(x2[grDevices::chull(x2),])
}

y<- x |> TDAbiplot(dist=1,inflate=1, alpha_Elip=HullPeeling, alpha=0.4)
```

Index

`bipl5`, 2

`colorpal`, 3

`FMbiplot`, 2, 3, 5, 10

`kable`, 7

`PCAbiplot`, 2, 4, 9, 10

`plot.bipl5`, 6

`plot_ly`, 4–6, 8–10

`predict.bipl5`, 2, 6

`princomp`, 4

`print.bipl5`, 2, 5, 7

`svd`, 5, 10

`Symbol_List`, 8

`TDAbiplot`, 2, 5, 8