Package 'biplotEZ'

November 13, 2024

Title EZ-to-Use Biplots

Version 2.2

Description Provides users with an EZ-to-use platform for representing data with biplots. Currently principal component analysis (PCA), canonical variate analysis (CVA) and simple correspondence analysis (CA) biplots are included. This is accompanied by various formatting options for the samples and axes. Alpha-bags and concentration ellipses are included for visual enhancements and interpretation. For an extensive discussion on the topic, see Gower, J.C., Lubbe, S. and le Roux, N.J. (2011, ISBN: 978-0-470-01255-0) Understanding Biplots. Wiley: Chichester.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

VignetteBuilder knitr

Config/testthat/edition 3

Suggests caret, cluster, geometry, ggplot2, ggrepel, grid, knitr, MASS, R.devices, rgl, rmarkdown, testthat (>= 3.0.0)

Imports graphics, grDevices, plotrix, splines, stats, withr

NeedsCompilation yes

Author Sugnet Lubbe [aut, cre, cph] (<https://orcid.org/0000-0003-2762-9944>), Niël le Roux [aut] (<https://orcid.org/0000-0002-1791-746X>), Johané Nienkemper-Swanepoel [aut] (<https://orcid.org/0000-0001-6086-8272>), Raeesa Ganey [aut] (<https://orcid.org/0009-0008-6973-0999>), Ruan Buys [aut] (<https://orcid.org/0000-0001-8527-8631>), Zoë-Mae Adams [aut] (<https://orcid.org/0000-0003-3730-4887>), Peter Manefeldt [aut] (<https://orcid.org/0000-0002-8604-843X>)

Maintainer Sugnet Lubbe <muvisu@sun.ac.za>

Repository CRAN

Date/Publication 2024-11-13 18:20:09 UTC

Contents

`alpha.bags` *Create alpha bags*

Description

This function produces α-bags, which is a useful graphical summary of the scatter plot. The alpha-bag refers to a contour which contains $\alpha\%$ of the observations.

Usage

```
alpha.bags(bp, alpha = 0.95, which = NULL, col = ez.col[which], lty = 1,
lwd = 1, max = 2500, trace = TRUE, opacity = 0.25, outlying=FALSE)
```

Arguments

`bp`	an object of class `biplot`.
`alpha`	numeric vector between 0 and 1 to determine coverage of the bag (α), with default `0.95`.
`which`	numeric vector indicating the selection of groups or classes to be fitted with α-bags.
`col`	vector of colours for the α-bags. Multiple α bags for one group will be displayed in the same colour.
`lty`	vector of line types for the α-bags. The same line type will be used per value of α.
`lwd`	vector of line widths for the α-bags. The same line width will be used per value of α.
`max`	maximum number of samples to include in α-bag calculations, with default 2500. If more samples are in the group, a random sample of size max is taken for the computations.
`trace`	logical, indicating progress of computation.
`opacity`	level of opacity, with default `0.5`.
`outlying`	logical indicating whether only outlying points should be plotted. Note the `which` argument may be overwritten when `TRUE`

Value

A list with the following components is available:

`alpha.bags`	list of coordinates for the α-bags for each group.
`col`	vector of colours for the α-bags.
`lty`	vector of line types for the α-bags.
`lwd`	vector of line widths for the α-bags.

References

Gower, J., Gardner-Lubbe, S. & Le Roux, N. (2011, ISBN: 978-0-470-01255-0) *Understanding Biplots.* Chichester, England: John Wiley & Sons Ltd.

Examples

```
biplot (iris[,1:4]) |> PCA(group.aes=iris[,5]) |> alpha.bags(alpha=0.95) |> plot()
biplot (iris[,1:4],group.aes=iris[,5]) |> PCA() |> alpha.bags(alpha=0.95) |> plot()
```

AoD *Use the Analysis of Distance (AoD) method to construct the biplot*

Description

This function appends the `biplot` object with elements resulting from using the AoD method.

Usage

```
AoD(bp, classes=bp$classes, dist.func=NULL, dist.func.cat=NULL,
dim.biplot = c(2,1,3), e.vects = 1:ncol(bp$X),
weighted = c("unweighted","weighted"), show.class.means = TRUE,
axes = c("regression","splines"), ...)
```

Arguments

`bp`	an object of class `biplot` obtained from preceding function `biplot()`.
`classes`	a vector of the same length as the number of rows in the data matrix with the class indicator for the samples.
`dist.func`	a character string indicating which distance function is used to compute the Euclidean embeddable distances between samples. One of `NULL` (default) which computes the Euclidean distance or other functions that can be used for the `dist()` function.
`dist.func.cat`	a character string indicating which distance function is used to compute the Euclidean embeddable distances between samples. One of `NULL` (default) which computes the extended matching coefficient or other functions.
`dim.biplot`	the dimension of the biplot. Only values `1`, `2` and `3` are accepted, with default `2`.
`e.vects`	the vector indicating which eigenvectors (canonical variates) should be plotted in the biplot, with default `1:dim.biplot`.
`weighted`	a character string indicating the weighting of the classes. One of `"unweighted"` for each class to receive equal weighting or `"weighted"` for each class to receive their class sizes as weights.
`show.class.means`	a logical value indicating whether to plot the class means on the biplot.
`axes`	a character string indicating the type of biplot axes to be used in the biplot. One of `"regression"` or `"splines"`.
`...`	more arguments to `dist.func`.

Value

Object of class `biplot`

Examples

```
biplot(iris[,1:4]) |> AoD(classes=iris[,5])
# create a CVA biplot
biplot(iris[,1:4]) |> AoD(classes=iris[,5]) |> plot()
```

`AoD.biplot` *Calculate elements for the Analysis of Distance (AoD) biplot*

Description

This function is used to construct the AoD biplot

Usage

```
## S3 method for class 'biplot'
AoD(
  bp,
  classes = bp$classes,
  dist.func = NULL,
  dist.func.cat = NULL,
  dim.biplot = c(2, 1, 3),
  e.vects = 1:ncol(bp$X),
  weighted = c("unweighted", "weighted"),
  show.class.means = TRUE,
  axes = c("regression", "splines"),
  ...
)
```

Arguments

`bp`	an object of class `biplot` obtained from preceding function `biplot()`.
`classes`	a vector of the same length as the number of rows in the data matrix with the class indicator for the samples.
`dist.func`	a character string indicating which distance function is used to compute the Euclidean embeddable distances between samples. One of `NULL` (default) which computes the Euclidean distance or other functions that can be used for the `dist()` function.
`dist.func.cat`	a character string indicating which distance function is used to compute the Euclidean embeddable distances between samples. One of `NULL` (default) which computes the extended matching coefficient or other functions.
`dim.biplot`	the dimension of the biplot. Only values `1`, `2` and `3` are accepted, with default `2`.

e.vects	the vector indicating which eigenvectors (canonical variates) should be plotted in the biplot, with default 1:dim.biplot.
weighted	a character string indicating the weighting of the classes. One of "unweighted" for each class to receive equal weighting or "weighted" for each class to receive their class sizes as weights.
show.class.means	a logical value indicating whether to plot the class means on the biplot.
axes	a character string indicating the type of biplot axes to be used in the biplot. One of "regression" or "splines".
...	more arguments to dist.func.

Value

an object of class biplot.

Examples

```
biplot(iris) |> AoD(classes = iris[,5]) |> plot()
```

---

axes *Format aesthetics for the biplot axes*

Description

This function allows the user to format the aesthetics for the biplot axes.

Usage

```
axes(bp, X.names=colnames(bp$X), which = 1:bp$p, col = grey(0.7), lwd = 1, lty = 1,
label.dir = "Orthog", label.col = col, label.cex = 0.75, label.line = 0.1,
label.offset=rep(0,4), ticks = 5, tick.col = col, tick.size = 1, tick.label = TRUE,
tick.label.side = "below", tick.label.col = tick.col, tick.label.cex = 0.6,
predict.col = col, predict.lwd = lwd, predict.lty = lty, ax.names = X.names,
orthogx = 0, orthogy = 0, vectors = FALSE, unit.circle=FALSE)
```

Arguments

bp	an object of class biplot.
X.names	a vector of column names of bp to specify which axes should be labelled.
which	a vector containing the columns or variables for which the axes should be displayed, with default 1:p.
col	the colour(s) for the axes, with default grey(0.7). Alternatively, provide a vector of colours corresponding to X.names.
lwd	the line width(s) for the axes, with default 1.

lty	the line type(s) for the axes, with default 1.
label.dir	a character string indicating the placement of the axis titles to the side of the figure. One of "Orthog" for axis titles to appear orthogonal to the side of the figure (default) , "Hor" for axis titles to appear horizontally or "Paral" for axis titles to appear parallel to the side of the figure.
label.col	the colour(s) for the axis labels, with default, col.
label.cex	the label expansion for the axis labels, with default 0.75.
label.line	the distance of the axis title from the side of the figure, with default 0.1.
label.offset	a four-component numeric vector controlling the distances axis titles are displayed from the side of the figure, with default rep(0,4). Sides are numbered 1 to 4 according to R conventions.
ticks	an integer-valued vector indicating the number of tickmarks for each axis, with default 5 for each axis.
tick.col	the colour(s) for the tick marks, with default col.
tick.size	a vector specifying the sizes of tick marks for each axis, with default 1 for each .
tick.label	a logical value indicating whether the axes should be labelled, with default TRUE.
tick.label.side	a character string indicating the position of the tick label. One of "below" for the label to appear below the tick mark (default) or "above" for the label to appear above the tick mark.
tick.label.col	the colour(s) for the tick mark labels, with default tick.col.
tick.label.cex	the label expansion for the tick mark labels, with default 0.6.
predict.col	the colour(s) for the predicted samples, with default col.
predict.lwd	the line width(s) for the predicted samples, with default lwd.
predict.lty	the line type(s) for the predicted samples, with default lty.
ax.names	a vector of size p containing user defined titles for the axes.
orthogx	a numeric vector of size p specifying the x-coordinate of the parallel transformation of each axis, with default 0 for each axis. This is only used when dim.biplot = 2.
orthogy	a numeric vector of size p specifying the y-coordinate of the parallel transformation of each axis, with default 0 for each axis. This is only used when dim.biplot = 2.
vectors	a logical value indicating whether vector representation (calibrated axes) should be displayed on the biplot. This is only used when a PCA biplot is produced.
unit.circle	a logical value indicating whether a unit circle should be displayed on the biplot.

Value

The object of class biplot will be appended with a list called axes containing the following elements:

which	a vector containing the columns for which the axes are displayed.

col	the colour(s) of the axes.
lwd	the line width(s) of the axes.
lty	the line type(s) of the axes.
label.dir	the placement of the axis titles to the side of the figure.
label.col	the colour(s) of the axis titles.
label.cex	the label expansion(s) of the axis titles.
label.line	the distance(s) of the axis titles from the side of the figure.
ticks	the number of tick marks per axis.
tick.col	the colour(s) of the tick marks.
tick.size	the size(s) of the tick marks.
tick.label	logical value(s) indicating whether axes are labelled.
tick.label.side	the position of the tick mark labels.
tick.label.col	the colour(s) of the tick mark labels.
tick.label.cex	the expansion(s) of the tick mark labels.
predict.col	the colour(s) of the predicted samples.
predict.lty	the line type(s) of the predicted samples.
predict.lwd	the line width(s) of the predicted samples.
names	the user defined axis titles.
orthogx	the horizontal translations for each axis.
orthogy	the vertical translations for each axis.
vectors	a logical value indicating whether calibrated axes are plotted.

See Also

biplot()

Examples

```
biplot(iris[,1:4]) |> PCA() |> axes(col="purple") |> plot()
biplot(iris[,1:4]) |> PCA() |> samples(col="purple",pch=15) |> axes() |> plot()
```

axes_coordinates *Calibrate Biplot Axes*

Description

Convenience function to obtain the coordinates of the calibrated ticks marks on the biplot

Usage

```
axes_coordinates(x)
```

Arguments

x	an object of class `biplot`

Value

An ordered list containing the coordinates the of tick marks to plotted on the biplot

Examples

```
x<-biplot(iris) |> PCA()
coordinates<-axes_coordinates(x)
```

biplot *First step to create a new biplot with* **biplotEZ**

Description

This function produces a list of elements to be used when producing a biplot, which provides a useful data analysis tool and allows the visual appraisal of the structure of large data matrices. Biplots are the multivariate analogue of scatter plots. They approximate the multivariate distribution of a sample in a few dimensions and they superimpose on this display representations of the variables on which the samples are measured.

Usage

```
biplot(data, classes = NULL, group.aes = NULL, center = TRUE, scaled = FALSE,
Title = NULL)
```

Arguments

data	a data frame or numeric matrix containing all variables the user wants to analyse.
classes	a vector identifying class membership.
group.aes	a vector identifying groups for aesthetic formatting.
center	a logical value indicating whether data should be column centered, with default TRUE.
scaled	a logical value indicating whether data should be standardised to unit column variances, with default FALSE.
Title	the title of the biplot to be rendered, enter text in " ".

Details

This function is the entry-level function in biplotEZ to construct a biplot display. It initialises an object of class biplot which can then be piped to various other functions to build up the biplot display.

Value

A list with the following components is available:

X	the matrix of the centered and scaled numeric variables.
Xcat	the data frame of the categorical variables.
raw.X	the original data.
classes	the vector of category levels for the class variable. This is to be used for colour, pch and cex specifications.
na.action	the vector of observations that have been removed.
center	a logical value indicating whether $\mathbf{X}$ is centered.
scaled	a logical value indicating whether $\mathbf{X}$ is scaled.
means	the vector of means for each numeric variable.
sd	the vector of standard deviations for each numeric variable.
n	the number of observations.
p	the number of variables.
group.aes	the vector of category levels for the grouping variable. This is to be used for colour, pch and cex specifications.
g.names	the descriptive names to be used for group labels.
g	the number of groups.
Title	the title of the biplot rendered

Useful links

The biplot display can be built up in four broad steps depending on the needs for the display. Firstly, choose an appropriate method to construct the display; Secondly, change the aesthetics of the display; Thirdly, append the display with supplementary features such as axes, samples and means; Finally, superimpose shapes, characters or elements onto the display.

1. Different types of biplots:

- `PCA()`: Principal Component Analysis biplot of various dimensions
- `CVA()`: Canonical Variate Analysis biplot
- `PCO()`: Principal Coordinate Analysis biplot
- `CA()`: Correspondence Analysis biplot
- `regress()`: Regression biplot method

2. Customise the biplot display with aesthetic functions:

- `samples()`: Change the formatting of sample points on the biplot display
- `axes()`: Change the formatting of the biplot axes

3. Supplement the existing biplot with additional axes, samples and group means:

- `newsamples()`: Add and change formatting of additional samples
- `newaxes()`: Add and change formatting of additional axes
- `means()`: Insert class means to the display, and format appropriately

4. Append the biplot display:

- `alpha.bags()`: Add α-bags
- `ellipses()`: Add ellipses
- `density2D()`: Add 2D density regions

Other useful links:

- `plot()`
- `fit.measures()`
- `legend.type()`
- `interpolate()`
- `prediction()`
- `classify()`
- `reflect()`
- `rotate()`

References

Gabriel, K.R. (1971) The biplot graphic display of matrices with application to principal component analysis. *Biometrika.* 58(3):453–467.

Gower, J., Gardner-Lubbe, S. & Le Roux, N. (2011, ISBN: 978-0-470-01255-0) *Understanding Biplots.* Chichester, England: John Wiley & Sons Ltd.

Gower, J.C. & Hand, D.J.(1996, ISBN: 0-412-71630-5) *Biplots.* London: Chapman & Hall.

Examples

```
biplot(data = iris)
# create a PCA biplot
biplot(data = iris) |> PCA() |> plot()
```

`biplotEZ` *biplotEZ: EZ-to-Use Biplots*

Description

Details

The goal of biplotEZ is to provide users an EZ-to-use platform for visually representing their data with biplots. Currently, this package includes principal component analysis (PCA) and canonical variate analysis (CVA) biplots. This is accompanied by various formatting options for the samples and axes. Alpha-bags and concentration ellipses are included for visual enhancements and interpretation.

Details

Package:	biplotEZ
Type:	Package
Version:	2.0
Date:	05-04-2024
License:	MIT
LazyLoad:	TRUE

Author(s)

- Sugnet Lubbe (Maintainer, muvisu@sun.ac.za)
- Niël le Roux
- Johané Nienkemper-Swanepoel
- Raeesa Ganey
- Ruan Buys
- Zoë-Mae Adams
- Peter Manefeldt

Core Functions

- biplot
- PCA
- CVA
- CA

Code Availability

The newest version of the package can be obtained on GitHub: https://github.com/MuViSU/biplotEZ

CA *Correspondence Analysis (CA) method*

Description

This function produces a list of elements to be used for CA biplot construction by approximation of the Pearson residuals.

Usage

```
CA(bp, dim.biplot = c(2,1,3), e.vects = 1:ncol(bp$X), variant = "Princ",
lambda.scal = FALSE)
```

Arguments

bp	object of class biplot obtained from preceding function biplot(center = FALSE). In order to maintain the frequency table, the input should not be centered or scaled. For CA, bp should be a contingency table.
dim.biplot	dimension of the biplot. Only values 1, 2 and 3 are accepted, with default 2.
e.vects	which eigenvectors (canonical variates) to extract, with default 1:dim.biplot.
variant	which correspondence analysis variant, with default "Princ", presents a biplot with rows in principal coordinates and columns in standard coordinates. variant = "Stand", presents a biplot with rows in standard coordinates and columns in principal coordinates. variant = "symmetric", presents a symmetric biplot with row and column standard coordinates scaled equally by the singular values.
lambda.scal	logical value to request lambda-scaling, default is FALSE. Controls stretching or shrinking of column and row distances.

Value

A list with the following components is available:

`Z`	Combined data frame of the row and column coordinates.
`r`	Numer of levels in the row factor.
`c`	Numer of levels in the column factor.
`Dr`	Diagonal matrix of row profiles.
`Dc`	Diagonal matrix of column profiles.
`Drh`	Weighted row profiles.
`Dch`	Weighted column profiles.
`rowcoor`	Row coordinates based on the selected `variant`.
`colcoor`	Column coordinates based on the selected `variant`.
`P`	Correspondence Matrix.
`Smat`	Standardised Pearson residuals.
`SVD`	Singular value decomposition solution: `d`, `u`, `v`.
`e.vects`	Depending on what was specified in `CA` argument.
`dim.biplot`	The dimension of the biplot.
`lambda.val`	The computed lambda value if lambda-scaling is requested.
`gamma`	Contribution of the singular values, based on the CA variant.

See Also

`biplot()`

Examples

```
# Creating a CA biplot with rows in principal coordinates:
biplot(HairEyeColor[,,2], center = FALSE) |> CA() |> plot()
# Creating a CA biplot with rows in standard coordinates:
biplot(HairEyeColor[,,2], center = FALSE) |> CA(variant = "Stand") |>
samples(col=c("magenta","purple"), pch = c(15,17), label.col = "black") |> plot()
# Creating a CA biplot with rows and columns scaled equally:
biplot(HairEyeColor[,,2], center = FALSE) |> CA(variant = "Symmetric") |>
samples(col = c("magenta","purple"), pch = c(15,17), label.col = "black") |> plot()
```

`CA.biplot` *CA biplot*

Description

Performs calculations for a CA biplot.

Usage

```
## S3 method for class 'biplot'
CA(
  bp,
  dim.biplot = c(2, 1, 3),
  e.vects = 1:ncol(bp$X),
  variant = "Princ",
  lambda.scal = FALSE
)
```

Arguments

`bp`	object of class `biplot` obtained from preceding function `biplot(center = FALSE)`. In order to maintain the frequency table, the input should not be centered or scaled. For `CA`, `bp` should be a contingency table.
`dim.biplot`	dimension of the biplot. Only values 1, 2 and 3 are accepted, with default `2`.
`e.vects`	which eigenvectors (canonical variates) to extract, with default `1:dim.biplot`.
`variant`	which correspondence analysis variant, with default "Princ", presents a biplot with rows in principal coordinates and columns in standard coordinates. `variant = "Stand"`, presents a biplot with rows in standard coordinates and columns in principal coordinates. `variant = "symmetric"`, presents a symmetric biplot with row and column standard coordinates scaled equally by the singular values.
`lambda.scal`	logical value to request lambda-scaling, default is `FALSE`. Controls stretching or shrinking of column and row distances.

Value

an object of class CA, inherits from class biplot.

Examples

```
biplot(HairEyeColor[,,2], center = FALSE) |> CA() |> plot()
```

CATPCA *Categorical Principal Component Analysis*

Description

Categorical Principal Component Analysis

Usage

```
CATPCA(bp, dim.biplot = c(2, 1, 3), e.vects = 1:ncol(bp$X),
group.aes = NULL, show.class.means = FALSE)
```

Arguments

bp	an object of class `biplot` obtained from preceding function `biplot()`.
dim.biplot	dimension of the biplot. Only values 1, 2 and 3 are accepted, with default 2.
e.vects	which eigenvectors (principal components) to extract, with default `1:dim.biplot`.
group.aes	vector of the same length as the number of rows in the data matrix for differentiated aesthetics for samples.
show.class.means	logical, indicating whether group means should be plotted in the biplot.

Value

an object of class `biplot`

Examples

```
CATPCA (iris)
```

classification *Classification biplot method*

Description

This function produces a list of elements to be used for constructing a classification biplot.

Usage

```
classification(bp, Pmat, dim.biplot = c(2, 1, 3), e.vects = 1:ncol(bp$X),
group.aes=NULL, axes = "regression", col=ez.col, opacity=0.4, borders = FALSE)
```

Arguments

bp	an object of class biplot obtained from preceding function biplot().
Pmat	a matrix containing the posterior probability for the classes
dim.biplot	dimension of the biplot. Only values 1, 2 and 3 are accepted, with default 2.
e.vects	which eigenvectors (principal components) to extract, with default 1:dim.biplot.
group.aes	vector of the same length as the number of rows in the data matrix for differentiated aesthetics for samples.
axes	type of axes, defaults to "regression"
col	colour of the classification regions
opacity	opacity of classification regions
borders	logical, indicating whether borders should be added to classification regions

Value

Object of class biplot with the following elements:

References

Gardner-Lubbe, S., 2016. A triplot for multiclass classification visualisation. *Computational Statistics & Data Analysis*, 94, pp.20-32.

Examples

```
biplot(iris[,1:4]) |>
classification(predict(MASS::lda(Species ~ ., data = iris))$posterior)
# create a classification biplot
biplot(iris[,1:4]) |>
classification(predict(MASS::lda(Species ~ ., data = iris))$posterior) |>
plot()
```

---

classification.biplot *classification biplot*

Description

Performs calculations for a classification biplot.

Usage

```
## S3 method for class 'biplot'
classification(
  bp,
  Pmat,
  dim.biplot = c(2, 1, 3),
  e.vects = 1:ncol(bp$X),
```

```
  group.aes = NULL,
  axes = "regression",
  col = ez.col,
  opacity = 0.4,
  borders = FALSE
)
```

Arguments

bp	an object of class biplot obtained from preceding function biplot().
Pmat	a matrix containing the posterior probability for the classes
dim.biplot	dimension of the biplot. Only values 1, 2 and 3 are accepted, with default 2.
e.vects	which eigenvectors (principal components) to extract, with default 1:dim.biplot.
group.aes	vector of the same length as the number of rows in the data matrix for differentiated aesthetics for samples.
axes	type of axes, defaults to "regression"
col	colour of the classification regions
opacity	opacity of classification regions
borders	logical, indicating whether borders should be added to classification regions

Value

an object of class biplot.

---

classify *Classify samples into classes*

Description

Classify samples into classes

Usage

```
classify(
  bp,
  classify.regions = TRUE,
  col = ez.col,
  opacity = 0.4,
  borders = FALSE
)
```

Arguments

`bp`	an object of class `biplot`
`classify.regions`	a logical value indicating whether classifications regions should be shown in the biplot, with default `TRUE`.
`col`	the colours of the classification regions
`opacity`	the opacity levels of the classification regions
`borders`	the border colours of the classification regions

Value

A list object called `classify` appended to the object of class `biplot` with the following elements:

`table`	the confusion matrix resulting from the classification into classes.
`rate`	the classification accuracy rate.
`classify.regions`	a logical value indicating whether classification regions are shown in the biplot.
`aes`	a list of chosen aesthetics for the colours, opacity levels and border colours of the classification regions.

Examples

```
biplot(iris[,1:4],classes = iris[,5]) |> CVA() |> axes(col="black") |>
  classify(col=c("red","blue","orange"),opacity=0.1) |> plot()
```

`CLPs` *Format aesthetics for the category level points*

Description

This function allows the user to format the aesthetics for the category level points (CLPs).

Usage

```
CLPs (bp,  which = 1:ncol(bp$Xcat), col = "black", cex = 0.6)
```

Arguments

`bp`	an object of class `biplot`.
`which`	a vector containing the columns or variables for which the CLPs should be displayed, with default `1:ncol(Xcat)`.
`col`	the colour(s) for the CLPs, with default `black`.
`cex`	the character expansion(s) for the CLPs, with default `0.6`.

Value

The object of class `biplot` will be appended with a list called `CLP.aes` containing the following elements A list with the following components is available:

`which`	a vector containing the columns or variables for which the CLPs are displayed.
`col`	the colour(s) of the CLPs.
`cex`	the character expansion(s) of the plotting characters of the CLPs.

See Also

`biplot`, `CA`, `AoD`

Examples

```
mtdf <- as.data.frame(mtcars)
mtdf$cyl <- factor(mtdf$cyl)
mtdf$vs <- factor(mtdf$vs)
mtdf$am <- factor(mtdf$am)
mtdf$gear <- factor(mtdf$gear)
mtdf$carb <- factor(mtdf$carb)
biplot(mtdf[,-11], scaled = TRUE) |> AoD(classes = mtdf[,11]) |>
CLPs(col = list(rep("olivedrab",3), rep("orange",2),
                rep("coral",2), rep("brown",3))) |>
plot()
```

CLRs *Format aesthetics for the category level regions*

Description

This function allows the user to format the aesthetics for the category level points (CLRs).

Usage

```
CLRs (bp,  which = 1, col = "black")
```

Arguments

`bp`	an object of class `biplot`.
`which`	the column name or number for which the CLRs should be displayed, with default `1`. Only one variable can be selected at a time.
`col`	the colours for the CLRs, with default `colorRampPalette(c("black","white"))`.

Value

The object of class `biplot` will be appended with a list called `CLP.aes` containing the following elements A list with the following components is available:

`which`	the variable number for which the CLRs are displayed.
`col`	the colours of the CLRs.

See Also

`biplot`, `PCO`, `AoD`

Examples

```
mtdf <- as.data.frame(mtcars)
mtdf$cyl <- factor(mtdf$cyl)
mtdf$vs <- factor(mtdf$vs)
mtdf$am <- factor(mtdf$am)
mtdf$gear <- factor(mtdf$gear)
mtdf$carb <- factor(mtdf$carb)
#biplot(mtdf[,-11], scaled = TRUE) |> PCO(group.aes = mtdf[,11]) |>
#CLRs(which = 10, col = "coral") |> plot()
```

CVA *Perform Canonical Variate Analysis (CVA)*

Description

This function appends the `biplot` object with elements resulting from performing CVA.

Usage

```
CVA(bp, classes=bp$classes, dim.biplot = c(2, 1, 3), e.vects = 1:ncol(bp$X),
           weightedCVA = "weighted", show.class.means = TRUE,
           low.dim = "sample.opt")
```

Arguments

`bp`	an object of class `biplot` obtained from preceding function `biplot()`.
`classes`	a vector of the same length as the number of rows in the data matrix with the class indicator for the samples.
`dim.biplot`	the dimension of the biplot. Only values `1`, `2` and `3` are accepted, with default `2`.
`e.vects`	the vector indicating which eigenvectors (canonical variates) should be plotted in the biplot, with default `1:dim.biplot`.

`weightedCVA` a character string indicating which type of CVA to perform. One of "`weighted`" (default) for a weighted CVA to be performed (The centring matrix will be a diagonal matrix with the class sizes ($\mathbf{C} = \mathbf{N}$), "`unweightedCent`" for unweighted CVA to be performed (The centring matrix is the usual centring matrix ($\mathbf{C} = \mathbf{I}_G - G^{-1}\mathbf{1}_G\mathbf{1}_G'$)) or "`unweightedI`" for unweighted CVA to be performed while retaining the weighted centroid (The centring matrix is an indicator matrix ($\mathbf{C} = \mathbf{I}_G$)).

`show.class.means` a logical value indicating whether to plot the class means on the biplot.

`low.dim` a character string indicating which method to use to construct additional dimension(s) if the dimension of the canonical space is smaller than `dim.biplot`. One of "`sample.opt`" (default) for maximising the sample predictivity of the individual samples in the biplot or "`Bhattacharyya.dist`" which is based on the decomposition of the Bhattacharyya distance into a component for the sample means and a component for the dissimilarity between the sample covariance matrices.

Value

Object of class CVA with the following elements:

`X` the matrix of the centered and scaled numeric variables.

`Xcat` the data frame of the categorical variables.

`raw.X` the original data.

`classes` the vector of category levels for the class variable. This is to be used for `colour`, `pch` and `cex` specifications.

`na.action` the vector of observations that have been removed.

`center` a logical value indicating whether $\mathbf{X}$ is centered.

`scaled` a logical value indicating whether $\mathbf{X}$ is scaled.

`means` the vector of means for each numerical variable.

`sd` the vector of standard deviations for each numerical variable.

`n` the number of observations.

`p` the number of variables.

`group.aes` the vector of category levels for the grouping variable. This is to be used for `colour`, `pch` and `cex` specifications.

`g.names` the descriptive names to be used for group labels.

`g` the number of groups.

`Title` the title of the biplot rendered.

`Lmat` the matrix for transformation to the canonical space.

`Linv` the inverse of $\mathbf{L}$.

`eigenvalues` the vector of eigenvalues of the two-sided eigenvalue problem.

`Z` the matrix with each row containing the details of the points to be plotted (i.e. coordinates).

ax.one.unit	one unit in the positive direction of each biplot axis.
Gmat	the indicator matrix defining membership of the classes.
Xmeans	the matrix of the class means.
Zmeans	the matrix of the class mean coordinates that are plotted in the biplot.
e.vects	the vector indicating which canonical variates are plotted in the biplot.
Cmat	the centring matrix based on different choices of weighting described in arguments.
Bmat	the between class sums of squares and cross products matrix.
Wmat	the within class sums of squares and cross products matrix.
Mrr	the matrix used for prediction from the canonical space (the inverse of $\mathbf{M} = \mathbf{LV}$).
Mr	the first r dimensions of the solution to be plotted.
Nmat	the matrix with the class sizes on the diagonal.
lambda.mat	the matrix with the eigenvalues of $\mathbf{W}^{-1/2}\mathbf{BW}^{-1/2}$ on the diagonal.
class.means	a logical value indicating whether the class means should be plotted in the biplot.
dim.biplot	the dimension of the biplot.
low.dim	the method used to construct additional dimension(s).

See Also

biplot()

Examples

```
biplot(iris[,1:4]) |> CVA(classes=iris[,5])
# create a CVA biplot
biplot(iris[,1:4]) |> CVA(classes=iris[,5]) |> plot()
```

CVA.biplot *Calculate elements for the CVA biplot*

Description

This function performs calculations for the construction of a CVA biplot.

Usage

```
## S3 method for class 'biplot'
CVA(
  bp,
  classes = bp$classes,
  dim.biplot = c(2, 1, 3),
  e.vects = 1:ncol(bp$X),
  weightedCVA = "weighted",
  show.class.means = TRUE,
  low.dim = "sample.opt"
)
```

Arguments

bp	an object of class biplot obtained from preceding function biplot().
classes	a vector of the same length as the number of rows in the data matrix with the class indicator for the samples.
dim.biplot	the dimension of the biplot. Only values 1, 2 and 3 are accepted, with default 2.
e.vects	the vector indicating which eigenvectors (canonical variates) should be plotted in the biplot, with default 1:dim.biplot.
weightedCVA	a character string indicating which type of CVA to perform. One of "weighted" (default) for a weighted CVA to be performed (The centring matrix will be a diagonal matrix with the class sizes ($\mathbf{C} = \mathbf{N}$), "unweightedCent" for unweighted CVA to be performed (The centring matrix is the usual centring matrix ($\mathbf{C} = \mathbf{I}_G - G^{-1}\mathbf{1}_G\mathbf{1}_G'$)) or "unweightedI" for unweighted CVA to be performed while retaining the weighted centroid (The centring matrix is an indicator matrix ($\mathbf{C} = \mathbf{I}_G$)).
show.class.means	a logical value indicating whether to plot the class means on the biplot.
low.dim	a character string indicating which method to use to construct additional dimension(s) if the dimension of the canonical space is smaller than dim.biplot. One of "sample.opt" (default) for maximising the sample predictivity of the individual samples in the biplot or "Bhattacharyya.dist" which is based on the decomposition of the Bhattacharyya distance into a component for the sample means and a component for the dissimilarity between the sample covariance matrices.

Value

an object of class CVA, inherits from class biplot.

Examples

```
biplot(iris[,1:4]) |> CVA(classes=iris[,5])
```

CVAlowdim *Construct additional dimensions when the dimension of the canonical space is smaller than the dimension of the biplot*

Description

This function is used to add dimensions to the CVA biplot when the dimension of the canonical space K is smaller than the dimension of the biplot (`dim.biplot`). This function is already used in the CVA calculations, and will therefore not have to be used in isolation.

Usage

```
CVAlowdim(bp, G, W, Mmat, low.dim, K, e.vects)
```

Arguments

`bp`	an object of class `biplot`.
`G`	the indicator matrix defining membership of the classes.
`W`	the within class sums of squares and cross products matrix.
`Mmat`	the eigenvector matrix from CVA.
`low.dim`	a character string indicating which method to use to construct additional dimension(s) if the dimension of the canonical space is smaller than `dim.biplot`. One of `"sample.opt"` (default) for maximising the sample predictivity of the individual samples in the biplot or `Bhattacharyya.dist` which is based on the decomposition of the Bhattacharyya distance into a component for the sample means and a component for the dissimilarity between the sample covariance matrices.
`K`	the dimension of the canonical space.
`e.vects`	the vector indicating which canonical variates are plotted in the biplot, with default `1:dim.biplot`

Value

A list with three components:

`Mr`	the first r dimensions of the solution to be plotted.
`Mrr`	the matrix used for prediction from the canonical space.
`Lmat`	the matrix for transformation to the canonical space.

`density1D` *Creates a kernel density in 1-dimension*

Description

Creates a kernel density in 1-dimension

Usage

```
density1D(
  bp,
  which = NULL,
  h = "nrd0",
  kernel = "gaussian",
  col = ez.col,
  lwd = 1.5,
  legend.mar = c(2, 5, 0, 5)
)
```

Arguments

`bp`	object of class `biplot`
`which`	which group.
`h`	bandwidth.
`kernel`	character string giving the smoothing kernel to be used.
`col`	colours to be used for each of the density curves.
`lwd`	linewidth of density curve.
`legend.mar`	The margin line of the legend.

Value

An object of class `biplot`.

Examples

```
biplot (iris,classes=iris[,5]) |> CVA(dim=1) |> density1D() |> plot()
```

`density2D` *Create a density in 2-dimensions*

Description

Create a density in 2-dimensions

Usage

```
density2D(
  bp,
  which = NULL,
  contours = F,
  h = NULL,
  n = 100,
  col = c("green", "yellow", "red"),
  contour.col = "black",
  cuts = 50,
  cex = 0.6,
  tcl = -0.2,
  mgp = c(0, -0.25, 0),
  layout.heights = c(100, 10),
  legend.mar = c(2, 5, 0, 5)
)
```

Arguments

`bp`	object of class `biplot`
`which`	which group to create a density; limited to only a single group at a time. If NULL, density drawn over all data points.
`contours`	logical indicating whether contours are added to the density plot
`h`	vector of bandwidths for x and y directions, see `kde2d`.
`n`	number of grid points in each direction. Can be scalar or a length-2 integer vector.
`col`	vector of colours to use to form a 'continuous' sequence of colours.
`contour.col`	colour of the contours.
`cuts`	number of colours in `col`.
`cex`	character expansion.
`tcl`	The length of tick marks as a fraction of the height of a line of text.
`mgp`	The margin line.
`layout.heights`	A vector of values for the heights of rows.
`legend.mar`	The margin line of the legend.

Value

An object of class `biplot`.

Examples

```
biplot(iris[,1:4],group.aes = iris[,5]) |> PCA() |>
  density2D(which=3,col=c("white","purple","cyan","blue")) |> plot()
biplot(iris[,1:4],group.aes = iris[,5]) |> PCA() |>
  density2D(which=3,col=c("white","purple","cyan","blue"),contours = TRUE,
  contour.col = "grey") |> plot()
```

---

ellipses *Concentration ellipses (κ-ellipses)*

Description

This function produces κ-ellipses, which is a useful geometrical description of the data points about the sample mean.

Usage

```
ellipses(bp, df=2, kappa = NULL, which = NULL,
alpha = 0.95, col = bp$sample$col[which], lty = 1, lwd = 1,
opacity = 0.25, trace = TRUE)
```

Arguments

`bp`	an object of class `biplot`.
`df`	degrees of freedom, with default `2`.
`kappa`	value to construct κ-ellipse (the value of κ).
`which`	the selection of the group for ellipse construction.
`alpha`	size of α-bag, with default `0.95`.
`col`	colour of ellipse. Multiple κ-ellipse for one group will be displayed in the same colour.
`lty`	line type of ellipse. The same line type will be used per value of κ.
`lwd`	line width of ellipse. The same line width will be used per value of κ.
`opacity`	level of opacity, with default `0.25`.
`trace`	logical, indicating progress of computation.

Value

A list with the following components is available:

`conc.ellipses`	list of coordinates for the κ-ellipses for each group.
`col`	vector of colours for the κ-ellipses.
`lty`	vector of line types for the κ-ellipses.
`lwd`	vector of line widths for the κ-ellipses.
`alpha`	vector of α values.

References

Gower, J., Gardner-Lubbe, S. & Le Roux, N. (2011, ISBN: 978-0-470-01255-0) *Understanding Biplots.* Chichester, England: John Wiley & Sons Ltd.

Examples

```
biplot (iris[,1:4]) |> PCA(group.aes=iris[,5]) |> ellipses(kappa=2) |> plot()
```

`extended.matching.coefficient` *Extended matching coefficient*

Description

Extended matching coefficient

Usage

```
extended.matching.coefficient(X)
```

Arguments

`X`	a data frame containing the categorical variables used for computing the EMC distance

Value

a dist object

Examples

```
mtdf <- as.data.frame(mtcars)
mtdf$cyl <- factor(mtdf$cyl)
mtdf$vs <- factor(mtdf$vs)
mtdf$am <- factor(mtdf$am)
mtdf$gear <- factor(mtdf$gear)
mtdf$carb <- factor(mtdf$carb)
extended.matching.coefficient(mtdf[,8:11])
```

---

fit.measures *Compute measures of fit for the biplot.*

Description

This function computes the measures of fit for the biplot. The biplot object is augmented with additional items, which can differ depending on the type of biplot. The measures provide information on the overall quality of fit and the adequacy of representation of variables.

Usage

```
fit.measures(bp)
```

Arguments

bp	an object of class biplot.

Value

An object of class biplot. The object is augmented with additional items, depending on the type of biplot object.

quality	the overall quality of fit.
adequacy	the adequacy of representation of variables.

For an object of class PCA:

axis.predictivity	the fit measure of each individual axis.
sample.predictivity	the fit measure for each individual sample.

For an object of class CVA:

axis.predictivity	the fit measure of each individual axis.
class.predictivity	the fit measure for each class mean.

`within.class.axis.predictivity`
: the fit measure for each axis based on values expressed as deviations from their class means.

`within.class.sample.predictivity`
: the fit measure for each sample expressed as deviation from its class mean.

For an object of class `CA`:

`row.predictivity`
: the fit measure for each row of the input matrix individual sample.

`col.predictivity`
: the fit measure for each column of the input matrix individual sample.

`Xhat`
: predicted matrix per row profile

Examples

```
out <- biplot (iris[,1:4]) |> PCA() |> fit.measures()
summary(out)
```

`interpolate` *Interpolate supplementary points and variables to add to the biplot*

Description

This function adds supplementary points and variables to the plot from a new data set.

Usage

```
interpolate(bp, newdata = NULL, newvariable = NULL)
```

Arguments

`bp`
: an object of class `biplot` obtained from preceding function `biplot()`.

`newdata`
: a new data set, similar in structure to the data set supplied to `biplot()` containing supplementary data points to be added onto the biplot.

`newvariable`
: a new data set, similar in structure to the data set supplied to `biplot()` containing supplementary variables to be added onto the biplot.

Value

The object of class `biplot` will be appended with the following elements:

`Xnew.raw`
: the new data.

`Xnew`
: the matrix of the centered and scaled new numeric variables of new data.

`Xnew.cat`
: the matrix of the categorical variables of new data.

`Znew`
: the matrix of the coordinates of the new data in the biplot.

For an object of class CA the following additional elements will be appended:

newrowcoor	the matrix of row coordinates of the new data in the biplot.
newcolcoor	the matrix of column coordinates of the new data in the biplot.

Examples

```
biplot(data = iris[1:145,]) |> PCA() |> interpolate(newdata = iris[146:150,]) |> plot()
biplot(HairEyeColor[,,2], center = FALSE) |> CA(variant = "Symmetric") |>
     interpolate(newdata = HairEyeColor[,,1]) |> plot()
```

---

legend.type *Format the legend for the biplot*

Description

This function enables the user to format the legend and make a required selection to display.

Usage

```
legend.type(bp, samples = FALSE, means = FALSE, bags = FALSE,
                   ellipses=FALSE,regions=FALSE, new=FALSE, ...)
```

Arguments

bp	an object of class biplot.
samples	a logical value indicating whether a legend should be printed for samples, with default FALSE.
means	a logical value indicating whether a legend should be printed for means, with default FALSE.
bags	a logical value indicating whether a legend should be printed for bags, with default FALSE.
ellipses	a logical value indicating whether a legend should be printed for concentration ellipses, with default FALSE.
regions	a logical value indicating whether a legend should be printed for classification regions, with default FALSE.
new	a logical value indicating whether the legend should appear in a new window, with default FALSE.
...	additional arguments to be sent to legend().

Value

A list with the following components is available:

samples	a logical value indicating whether a legend for samples are provided.
means	a logical value indicating whether a legend for class means are provided.
bags	a logical value indicating whether a legend for α-bags are provided.
ellipses	a logical value indicating whether a legend for κ-ellipses are provided.
regions	a logical value indicating whether a legend for classification regions are provided.
new	a logical value indicating whether the legend appears on new plot.

Examples

```
biplot (iris[,1:4], Title="Test biplot") |> PCA(group.aes = iris[,5]) |>
    legend.type(samples=TRUE) |> plot()
```

---

means *Format aesthetics for the class or group means*

Description

This function allows the user to format the aesthetics for the class means or group means.

Usage

```
means (bp,  which = NULL, col = NULL, pch = 15, cex = 1, label = FALSE,
label.col = NULL,label.cex = 0.75, label.side = "bottom", label.offset = 0.5,
opacity = 1, shade.darker = TRUE)
```

Arguments

bp	an object of class biplot.
which	a vector containing the groups or classes for which the means should be displayed, with default bp$g.
col	the colour(s) for the means, with default as the colour of the samples.
pch	the plotting character(s) for the means, with default 15.
cex	the character expansion(s) for the means, with default 1.
label	a logical value indicating whether the means should be labelled, with default TRUE.
label.col	a vector of the same length as which with label colours for the means, with default as the colour of the means.
label.cex	a vector of the same length as which with label text expansions for the means, with default 0.75.

label.side	the side at which the label of the plotted mean point appears, with default bottom. Note that unlike the argument pos in text(), options are "bottom", "left", "top", "right" and not 1, 2, 3, 4.
label.offset	the offset of the label from the plotted mean point. See ?text for a detailed explanation of the argument offset.
opacity	transparency of means.
shade.darker	a logical value indicating whether the colour of the mean points should be made a shade darker than the default or specified colour, with default TRUE.

Details

The number of classes or groups (defined by group.aes) is indicated as g. If an argument is not of length g, recycling is used.

Value

The object of class biplot will be appended with a list called means containing the following elements:

which	a vector containing the groups or classes for which the means are displayed.
col	the colour(s) of the means.
pch	the plotting character(s) of the means.
cex	the character expansion(s) of the plotting character(s) of the means.
label	a logical value indicating whether means are labelled.
label.col	the label colours of the means.
label.cex	the label text expansions of the samples.
label.side	the side at which the label of the plotted mean point appears.
label.offset	the offset of the label from the plotted mean point.
opacity	the opacity level of the plotted points.

See Also

biplot()

Examples

```
biplot(iris[,1:4]) |> PCA() |>
          means(col = "purple", pch = 15, cex = 2) |> plot()
```

newaxes *Format aesthetics for the supplementary (new) biplot axes*

Description

This function allows the user to format the aesthetics for the supplementary (new) biplot axes.

Usage

```
newaxes(bp, X.new.names=bp$var.names, which = 1:bp$num.vars, col = "orange", lwd = 1,
lty = 1, label.dir = "Orthog", label.col = col, label.cex = 0.75, label.line = 0.1,
ticks = 5, tick.col = col, tick.size = 1, tick.label = TRUE, tick.label.col = tick.col,
tick.label.cex = 0.6, tick.label.side = "below", predict.col = col, predict.lwd = lwd,
predict.lty = lty, ax.names = X.new.names, orthogx = 0, orthogy = 0)
```

Arguments

bp	an object of class biplot.
X.new.names	a vector of the new column names of bp to specify which axes should be labelled.
which	a vector containing the new columns or variables for which the axes should be displayed, with default 1:num.vars.
col	the colour(s) for the axes, with default grey(0.7). Alternatively, provide a vector of colours corresponding to X.names.
lwd	the line width(s) for the axes, with default 1.
lty	the line type(s) for the axes, with default 1.
label.dir	a character string indicating the placement of the axis titles to the side of the figure. One of "Orthog" for axis titles to appear orthogonal to the side of the figure (default) , "Hor" for axis titles to appear horizontally or "Paral" for axis titles to appear parallel to the side of the figure.
label.col	the colour(s) for the axis labels, with default, col.
label.cex	the label expansion for the axis labels, with default 0.75.
label.line	the distance of the axis title from the side of the figure, with default 0.1.
ticks	an integer-valued vector indicating the number of tickmarks for each axis, with default 5 for each axis.
tick.col	the colour(s) for the tick marks, with default col.
tick.size	a vector specifying the sizes of tick marks for each axis, with default 1 for each .
tick.label	a logical value indicating whether the axes should be labelled, with default TRUE.
tick.label.col	the colour(s) for the tick mark labels, with default tick.col.
tick.label.cex	the label expansion for the tick mark labels, with default 0.6.

`tick.label.side`
: a character string indicating the position of the tick label. One of "`below`" for the label to appear below the tick mark (default) or "`above`" for the label to appear above the tick mark.

`predict.col`
: the colour(s) for the predicted samples, with default `col`.

`predict.lwd`
: the line width(s) for the predicted samples, with default `lwd`.

`predict.lty`
: the line type(s) for the predicted samples, with default `lty`.

`ax.names`
: a vector of size `p` containing user defined titles for the axes.

`orthogx`
: a numeric vector of size `p` specifying the x-coordinate of the parallel transformation of each axis, with default `0` for each axis. This is only used when `dim.biplot = 2`.

`orthogy`
: a numeric vector of size `p` specifying the y-coordinate of the parallel transformation of each axis, with default `0` for each axis. This is only used when `dim.biplot = 2`.

Value

The object of class `biplot` will be appended with a list called `newaxes` containing elements similar to that of `axes`.

See Also

`biplot`, `axes`

Examples

```
biplot(data = iris[,1:2]) |> PCA() |> interpolate(newvariable = iris[3:4]) |>
  newaxes(col="gold") |> plot()
```

---

newsamples *Format aesthetics for the supplementary (new) biplot samples*

Description

This function allows formatting changes to new samples.

Usage

```
newsamples (bp,  col = "darkorange1", pch = 1, cex = 1, label = FALSE,
label.name = NULL, label.col = NULL,label.cex = 0.75, label.side = "bottom",
label.offset = 0.5, connected = FALSE, connect.col = "black", connect.lty=1,
connect.lwd=1)
```

Arguments

bp	an object of class biplot.
col	the colour(s) for the new samples, with default darkorange1.
pch	the plotting character(s) for the new samples, with default 1.
cex	the character expansion(s) for the new samples, with default 1.
label	a logical value indicating whether new samples should be labelled or not, with default FALSE.
label.name	the label names for the new samples.
label.col	a vector of the same length as the number of new samples containing the colour(s) for the labels of the new samples, with default the colour of the sample points.
label.cex	the label text expansion(s) for the new samples, with default 0.75.
label.side	the side at which the label of the plotted point appears, with default bottom. Note that unlike the argument pos in text(), options are "bottom", "left", "top","right" and not 1, 2, 3, 4.
label.offset	the offset of the label from the plotted point. See ?text for a detailed explanation of the argument offset.
connected	a logical value indicating whether samples are connected in order of rows of the data matrix, with default FALSE.
connect.col	the colour of the connecting line, with default black.
connect.lty	the line type of the connecting line, with default 1.
connect.lwd	the line width of the connecting line, with default 1.

Value

The object of class biplot will be appended with a list called newsamples containing the following elements:

col	the colour(s) of the new samples.
pch	the plotting character(s) of the new samples.
cex	the character expansion(s) of the plotting character(s) of the new samples.
label	a logical value indicating whether new samples are labelled.
label.col	the label colours of the new samples.
label.cex	the label text expansions of the new samples.
label.side	the side at which the label of the plotted point appears.
label.offset	the offset of the label from the plotted point.
connected	a logical value indicating whether new samples are connected.
connect.col	the colour of the connecting line.
connect.lty	the line type of the connecting line.
connect.lwd	the line width of the connecting line.

See Also

biplot, samples

Examples

```
biplot(data = iris[1:145,]) |> PCA() |> samples(col = "grey") |>
interpolate(newdata = iris[146:150,]) |> newsamples(col = rainbow(6), pch=15) |> plot()
```

PCA *Perform Principal Components Analysis (PCA)*

Description

This function appends the `biplot` object with elements resulting from performing PCA.

Usage

```
PCA(bp, dim.biplot = c(2, 1, 3), e.vects = 1:ncol(bp$X),
group.aes = NULL, show.class.means = FALSE, correlation.biplot = FALSE)
```

Arguments

`bp`	an object of class `biplot` obtained from preceding function `biplot()`.
`dim.biplot`	the dimension of the biplot. Only values `1`, `2` and `3` are accepted, with default `2`.
`e.vects`	the vector indicating which eigenvectors (principal components) should be plotted in the biplot, with default `1:dim.biplot`.
`group.aes`	a vector of the same length as the number of rows in the data matrix for differentiated aesthetics for samples.
`show.class.means`	a logical value indicating whether group means should be plotted in the biplot.
`correlation.biplot`	a logical value. If `FALSE`, the distances between sample points are optimally approximated in the biplot. If `TRUE`, the correlations between variables are optimally approximated by the cosine of the angles between axes. Default is `FALSE`.

Value

An object of class PCA with the following elements:

`X`	the matrix of the centered and scaled numeric variables.
`Xcat`	the data frame of the categorical variables.
`raw.X`	the original data.
`classes`	the vector of category levels for the class variable. This is to be used for `colour`, `pch` and `cex` specifications.

na.action	the vector of observations that have been removed.
center	a logical value indicating whether $\mathbf{X}$ is centered.
scaled	a logical value indicating whether $\mathbf{X}$ is scaled.
means	the vector of means for each numerical variable.
sd	the vector of standard deviations for each numerical variable.
n	the number of observations.
p	the number of variables.
group.aes	the vector of category levels for the grouping variable. This is to be used for colour, pch and cex specification.
g.names	the descriptive names to be used for group labels.
g	the number of groups.
Title	the title of the biplot rendered.
Z	the matrix with each row containing the details of the points that are plotted (i.e. coordinates).
Lmat	the matrix for transformation to the principal components.
Linv	the inverse of $\mathbf{L}$.
eigenvalues	the vector of eigenvalues of the covariance matrix of $\mathbf{X}$.
ax.one.unit	one unit in the positive direction of each biplot axis.
e.vects	the vector indicating which principal components are plotted in the biplot.
Vr	the 1:dim.biplot columns of $\mathbf{V}$.
dim.biplot	the dimension of the biplot.
class.means	a logical value indicating whether group means are plotted in the biplot.
Zmeans	the matrix of class mean coordinates that are plotted in the biplot.

References

Gabriel, K.R. (1971) The biplot graphic display of matrices with application to principal component analysis. *Biometrika.* 58(3):453–467.

See Also

biplot()

Examples

```
biplot(iris[,1:4]) |> PCA()
# create a PCA biplot
biplot(data = iris) |> PCA() |> plot()
```

PCA.biplot *Calculate elements for the PCA biplot*

Description

This function performs calculations for the construction of a PCA biplot.

Usage

```
## S3 method for class 'biplot'
PCA(
  bp,
  dim.biplot = c(2, 1, 3),
  e.vects = 1:ncol(bp$X),
  group.aes = NULL,
  show.class.means = FALSE,
  correlation.biplot = FALSE
)
```

Arguments

bp	an object of class biplot obtained from preceding function biplot().
dim.biplot	the dimension of the biplot. Only values 1, 2 and 3 are accepted, with default 2.
e.vects	the vector indicating which eigenvectors (principal components) should be plotted in the biplot, with default 1:dim.biplot.
group.aes	a vector of the same length as the number of rows in the data matrix for differentiated aesthetics for samples.
show.class.means	a logical value indicating whether group means should be plotted in the biplot.
correlation.biplot	a logical value. If FALSE, the distances between sample points are optimally approximated in the biplot. If TRUE, the correlations between variables are optimally approximated by the cosine of the angles between axes. Default is FALSE.

Value

an object of class PCA, inherits from class biplot.

Examples

```
biplot(iris[,1:4]) |> PCA()
# create a PCA biplot
biplot(data = iris) |> PCA() |> plot()
```

PCO *Principal Coordinate Analysis (PCO) biplot method*

Description

Principal Coordinate Analysis (PCO) biplot method

Usage

```
PCO(bp, Dmat=NULL, dist.func=NULL, dist.func.cat=NULL,
           dim.biplot = c(2,1,3), e.vects = NULL, group.aes=NULL,
           show.class.means = FALSE, axes = c("regression","splines"), ...)
```

Arguments

bp	an object of class `biplot` obtained from preceding function `biplot()`.
Dmat	nxn matrix of Euclidean embeddable distances between samples
dist.func	function to compute Euclidean embeddable distances between samples. The default NULL computes Euclidean distance.
dist.func.cat	function to compute Euclidean embeddable distance between categorical variables for the samples. The default NULL computes the extended matching coefficient.
dim.biplot	dimension of the biplot. Only values 1, 2 and 3 are accepted, with default 2.
e.vects	e.vects which eigenvectors (canonical variates) to extract, with default `1:dim.biplot`.
group.aes	vector of the same length as the number of rows in the data matrix for differentiated aesthetics for samples.
show.class.means	logical, indicating whether to plot the class means on the biplot.
axes	type of biplot axes, currently only regression axes are implemented
...	more arguments to `dist.func`

Value

Object of class biplot

Examples

```
biplot(iris[,1:4]) |> PCO(dist.func = sqrtManhattan)
# create a CVA biplot
biplot(iris[,1:4]) |> PCO(dist.func = sqrtManhattan) |> plot()
```

PCO.biplot *PCO biplot*

Description

Computes Principal Coordinate Analysis biplot

Usage

```
## S3 method for class 'biplot'
PCO(
  bp,
  Dmat = NULL,
  dist.func = NULL,
  dist.func.cat = NULL,
  dim.biplot = c(2, 1, 3),
  e.vects = NULL,
  group.aes = NULL,
  show.class.means = FALSE,
  axes = c("regression", "splines"),
  ...
)
```

Arguments

bp	an object of class biplot obtained from preceding function biplot().
Dmat	nxn matrix of Euclidean embeddable distances between samples
dist.func	function to compute Euclidean embeddable distances between samples. The default NULL computes Euclidean distance.
dist.func.cat	function to compute Euclidean embeddable distance between categorical variables for the samples. The default NULL computes the extended matching coefficient.
dim.biplot	dimension of the biplot. Only values 1, 2 and 3 are accepted, with default 2.
e.vects	e.vects which eigenvectors (canonical variates) to extract, with default 1:dim.biplot.
group.aes	vector of the same length as the number of rows in the data matrix for differentiated aesthetics for samples.
show.class.means	logical, indicating whether to plot the class means on the biplot.
axes	type of biplot axes, currently only regression axes are implemented
...	more arguments to dist.func

Value

an object of class biplot.

Examples

```
biplot(iris) |> PCO(dist.func=sqrtManhattan) |> plot()
```

`plot.biplot` *Generic Plotting function of objects of class biplot*

Description

Generic Plotting function of objects of class biplot

Usage

```
## S3 method for class 'biplot'
plot(
  x,
  exp.factor = 1.2,
  axis.predictivity = NULL,
  sample.predictivity = NULL,
  zoom = FALSE,
  xlim = NULL,
  ylim = NULL,
  ...
)
```

Arguments

`x`	An object of class `biplot`.
`exp.factor`	a numeric value with default axes of the biplot. Larger values are specified for zooming out with respect to sample points in the biplot display and smaller values are specified for zooming in with respect to sample points in the biplot display.
`axis.predictivity`	either a logical or a numeric value between `0` and `1`. If it is a numeric value, this value is used as threshold so that only axes with axis predictivity larger than the threshold is displayed. If `axis.predictivity = TRUE`, the axis colour is 'diluted' in proportion with the axis predictivity.
`sample.predictivity`	either a logical or a numeric value between 0 and 1. If it is a numeric value, this value is used as threshold so that only samples with sample predictivity larger than the threshold is displayed. If `sample.predictivity = TRUE`, the sample size is shrinked in proportion with the sample predictivity.
`zoom`	a logical value allowing the user to select an area to zoom into.
`xlim`	the horizontal limits of the plot.
`ylim`	the vertical limits of the plot.
`...`	additional arguments.

Value

An object of class `biplot`.

Examples

```
biplot (iris[,1:4]) |> PCA() |> plot()
```

`prediction` *Predict samples to display on the biplot*

Description

This function makes predictions of sample points, variables and means and displays them on the biplot.

Usage

```
prediction(bp, predict.samples = NULL, predict.means = NULL, which = 1:bp$p)
```

Arguments

`bp`	an object of class `biplot` obtained from preceding function `biplot()`.
`predict.samples`	a vector specifying which samples to predict.
`predict.means`	a vector specifying which group means to predict.
`which`	a vector specifying which variable to do the prediction.

Value

A list object called `predict` appended to the object of class `biplot` with the following elements:

`samples`	a vector of indices of samples which are being predicted.
`predict.means`	a vector of group names of groups for which the means are being predicted.
`which`	the vector of indices variables which are being predicted.
`predict.mat`	the matrix of predicted samples.
`predict.means.mat`	the matrix of predicted group means.

Examples

```
biplot(data = iris[,1:4]) |> PCA(group.aes=iris[,5], show.class.means = TRUE) |>
prediction(141:145,1:3) |> plot()
```

print.biplot *Generic print function for objects of class biplot*

Description

This function is used to print output when the biplot object is created.

Usage

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

Arguments

x	an object of class biplot.
...	additional arguments.

Value

This function will not produce a return value, it is called for side effects.

Examples

```
out <- biplot (iris[,1:4]) |> PCA()
out
```

reflect *Reflect the biplot about a chosen axis*

Description

This function provides the user with an option to reflect the biplot horizontally, vertically or diagonally.

Usage

```
reflect(bp, reflect.axis = c("FALSE", "x", "y", "xy"))
```

Arguments

bp	an object of class biplot
reflect.axis	a character string indicating which axis about to reflect. One of FALSE (default), "x" for reflection about the x-axis, "y" for reflection about the y-axis and "xy" for reflection about both axes.

Value

An object of class `biplot`

Examples

```
biplot(iris[,1:4],group.aes = iris[,5]) |> PCA() |> reflect("x") |> plot()
biplot(iris[,1:4],group.aes = iris[,5]) |> PCA() |> reflect("y") |> plot()
biplot(iris[,1:4],group.aes = iris[,5]) |> PCA() |> reflect("xy") |> plot()
```

---

regress *Regression biplot method*

Description

Regression biplot method

Usage

```
regress(bp, Z, group.aes=NULL, show.group.means = TRUE,
               axes = c("regression", "splines"))
```

Arguments

`bp`	an object of class `biplot` obtained from preceding function `biplot()`.
`Z`	the matrix of coordinates of the samples
`group.aes`	vector of the same length as the number of rows in the data matrix for differentiated aesthetics for samples.
`show.group.means`	logical, indicating whether group means should be plotted in the biplot.
`axes`	the type of axes to be fitted to the biplot. Options are 'regression' for linear regression axes (default) and 'splines' for B-spline axes.

Value

Object of class biplot

Examples

```
biplot(iris[,1:4]) |> regress(Z=cmdscale(dist(iris[,1:4]))) |> plot()
```

`regress.biplot` *Regression biplot*

Description

Computes regression biplot axes

Usage

```
## S3 method for class 'biplot'
regress(
  bp,
  Z,
  group.aes = NULL,
  show.group.means = TRUE,
  axes = c("regression", "splines")
)
```

Arguments

`bp`	an object of class `biplot` obtained from preceding function `biplot()`.
`Z`	the matrix of coordinates of the samples
`group.aes`	vector of the same length as the number of rows in the data matrix for differentiated aesthetics for samples.
`show.group.means`	logical, indicating whether group means should be plotted in the biplot.
`axes`	the type of axes to be fitted to the biplot. Options are 'regression' for linear regression axes (default) and 'splines' for B-spline axes.

Value

an object of class biplot.

Examples

```
biplot(iris) |> regress(Z = cmdscale(dist(iris[,1:4]))) |> plot()
```

rotate *Rotate the biplot a chosen amount of degrees*

Description

This function provides the user with an option to rotate the biplot anti-clockwise or clockwise.

Usage

```
rotate(bp, rotate.degrees = 0)
```

Arguments

bp	an object of class `biplot`
rotate.degrees	a value specifying the degrees the biplot should be rotated, with default `0`. A positive value results in anti-clockwise rotation and a negative value in clockwise rotation.

Value

An object of class `biplot`.

Examples

```
biplot(iris[,1:4],group.aes = iris[,5]) |> PCA() |> rotate(200) |> plot()
```

samples *Format aesthetics for the biplot samples*

Description

This function allows the user to format the aesthetics for the samples.

Usage

```
samples (bp,  which = 1:bp$g, col = ez.col, pch = 16, cex = 1,
label = FALSE, label.name = NULL, label.col=NULL, label.cex = 0.75,
label.side = "bottom", label.offset = 0.5,
connected=FALSE, connect.col = "black", connect.lty = 1,
connect.lwd = 1, opacity = 1)
```

Arguments

bp	an object of class biplot.
which	a vector containing the groups or classes for which the samples should be displayed, with default bp$g.
col	the colour(s) for the samples, with default blue.
pch	the plotting character(s) for the samples, with default 16.
cex	the character expansion(s) for the samples, with default 1.
label	a logical value indicating whether the samples should be labelled, with default FALSE. Alternatively, specify "ggrepel" for non-overlapping placement of labels.
label.name	a vector of the same length as which with label names for the samples, with default NULL. If NULL, the rownames(bp) are used. Alternatively, a custom vector of length n should be used.
label.col	a vector of the same length as which with label colours for the samples, with default as the same colour of the sample points.
label.cex	a vector of the same length as which with label text expansions for the samples, with default 0.75.
label.side	the side at which the label of the plotted point appears, with default bottom. Note that unlike the argument pos in text(), options are "bottom", "left", "top", "right" and not 1, 2, 3, 4.
label.offset	the offset of the label from the plotted point. See ?text for a detailed explanation of the argument offset.
connected	a logical value indicating whether samples are connected in order of rows of the data matrix, with default FALSE.
connect.col	the colour of the connecting line, with default black.
connect.lty	the line type of the connecting line, with default 1.
connect.lwd	the line width of the connecting line, with default 1.
opacity	the opacity level of the plotted points, with default 1 for an opaque point.

Details

The arguments which, col, pch and cex are based on the specification of group.aes or classes. If no groups are specified, a single colour, plotting character and / or character expansion is expected. If g groups are specified, vectors of length g is expected, or values are recycled to length g.

The arguments label, label.cex, label.side and label.offset are based on the sample size n. A single value will be recycled n times or a vector of length n is expected.

Value

The object of class biplot will be appended with a list called samples containing the following elements:

which	a vector containing the groups or classes for which the samples (and means) are displayed.

col	the colour(s) of the samples.
pch	the plotting character(s) of the samples.
cex	the character expansion(s) of the plotting character(s) of the samples.
label	a logical value indicating whether samples are labelled.
label.name	the label names of the samples.
label.col	the label colours of the samples.
label.cex	the label text expansions of the samples.
label.side	the side at which the label of the plotted point appears..
label.offset	the offset of the label from the plotted point.
connected	a logical value indicating whether samples are connected in order of the rows of the data matrix.
connect.col	the colour of the connecting line.
connect.lty	the line type of the connecting line.
connect.lwd	the line width of the connecting line.
opacity	the opacity level of the plotted points.

See Also

biplot()

Examples

```
biplot(iris[,1:4]) |> PCA() |> samples(col="purple",pch=15, opacity=0.5) |> plot()
biplot(iris[,1:4]) |> PCA() |>
  samples(col="purple",pch=NA, opacity=0.5, label = TRUE) |> plot()
biplot(iris[,1:4]) |> PCA() |>
  samples(col="purple",pch=NA, opacity=0.5, label = TRUE,
          label.name = paste("s:",1:150, sep="")) |>
  plot()
biplot(iris[,1:4]) |> PCA() |>
  samples(col="purple",pch=NA, opacity=0.5, label = "ggrepel") |> plot()
```

---

sqrtManhattan *Computes the square root of the Manhattan distance An example of a Euclidean embeddable distance metric*

Description

Computes the square root of the Manhattan distance An example of a Euclidean embeddable distance metric

Usage

```
sqrtManhattan(X)
```

Arguments

X	matrix of samples x variables for computation of samples x samples distance matrix

Value

a dist object

Examples

```
sqrtManhattan(iris[,1:4])
```

---

summary.biplot *Generic summary function for objects of class biplot*

Description

This function is used to print summary output of the biplot. These summary outputs are related to measures of fit.

Usage

```
## S3 method for class 'biplot'
summary(
  object,
  adequacy = TRUE,
  axis.predictivity = TRUE,
  sample.predictivity = TRUE,
  class.predictivity = TRUE,
  within.class.axis.predictivity = TRUE,
  within.class.sample.predictivity = TRUE,
  ...
)
```

Arguments

object	an object of class biplot.
adequacy	a logical value indicating whether variable adequacies should be reported, with default TRUE.
axis.predictivity	a logical value indicating whether axis predictivities should be reported, with default TRUE.
sample.predictivity	a logical value indicating whether sample predictivities should be reported, with default TRUE.

`class.predictivity`
: a logical value indicating whether class predictivities should be reported, with default `TRUE` (only applicable to objects of class `CVA`).

`within.class.axis.predictivity`
: a logical value indicating whether within class axis predictivity should be reported, with default `TRUE` (only applicable to objects of class `CVA`).

`within.class.sample.predictivity`
: a logical value indicating whether within class sample predictivity should be reported, with default `TRUE` (only applicable to objects of class `CVA`).

`...`
: additional arguments.

Value

This function will not produce a return value, it is called for side effects.

Examples

```
out <- biplot (iris[,1:4]) |> PCA() |> fit.measures()
summary(out)
```

---

translate_axes *Translate biplot axes*

Description

Automatically or manually translate the axes away from the center of the plot

Usage

```
translate_axes(bp, delta = 0, swop = FALSE, distances = NULL)
```

Arguments

`bp`	An object of class `biplot`
`delta`	numeric value indicating distance between axes
`swop`	logical. Change the direction in which axes are translated
`distances`	numeric vector of distances. Used to manually parallel translate the axes.

Details

This function uses the same algorithm implemented in `TDAbiplot` in the `bipl5` package. It translates the axes out of the center of the plot. Correlated axes generally gets translated in the same direction.

This function calculates the `orthogx` and `orthogy` paramaters in `axes()`

Value

An object of class `biplot` with the translated distances appended under bp$axes

Examples

```
#Translate the axes out of the plot center

bp <- biplot(state.x77,scaled = TRUE)|>
      CVA(state.region) |>
      translate_axes(swop=TRUE,delta =0.2)|>
      plot(exp.factor=3)

#adjust the distance of an axis

dist <- bp$axes$translate_distance
dist[7] <- 0.4
bp |> translate_axes(delta = 0.2, distances=dist) |> plot()
```

Index