

Package: bipl5 (via r-universe)

October 17, 2026

Title Construct Reactive Calibrated Axes Biplots

Version 1.1.0

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

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports biplotEZ (>= 2.0), cluster, crayon (>= 1.5.2), ggplot2 (>= 3.4.0), grid, htmltools, htmlwidgets (>= 1.6.2), knitr (>= 1.43), plotly (>= 4.10.2)

Depends R (>= 2.10)

URL <https://www.bipl5.co.za>, <https://github.com/RuanBuys/bipl5>

BugReports <https://github.com/RuanBuys/bipl5/issues>

Suggests pkgload, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libssl-dev

Repository <https://ruanbuys.r-universe.dev>

Date/Publication 2026-08-18 12:41:57 UTC

RemoteUrl <https://github.com/ruanbuys/bipl5>

RemoteRef HEAD

RemoteSha b5e36b4ad710c70c41dad80d9729e266ab7e6432

Contents

append_mdsDisplay	2
bipl5	3
colorpal	4
extract	4
format_samples	5
geom_calibrated_axis	9
init_biplot	12
overlay_fit	13
plot.bipl5_biplot	14
plot.bipl5_fit	14
print.bipl5_biplot	15
print.bipl5_data	16
print.bipl5_fitmeasures	16
print.bipl5_mdsDisplay	17
remove_mdsDisplay	17
scale_mds	18
score_axes	19
Symbol_List	20
wrap_bipl5	20
wrap_bipl5.CVA	21
wrap_bipl5.PCA	21
wrap_bipl5.PCO	28
wrap_bipl5.regress	29
wrap_bipl5_gg	34
Index	39

append_mdsDisplay	<i>Append a mdsDisplay to a bipl5_biplot object</i>
-------------------	---

Description

Adds a new biplot layer for a specified pair of principal components. The pair is sorted automatically (e.g. c(5, 3) becomes c(3, 5)). Both PC indices must be between 1 and p (the number of variables), and the pair must not already exist.

Usage

```
append_mdsDisplay(object, eigenvectors)

## S3 method for class 'bipl5_biplot'
append_mdsDisplay(object, eigenvectors)
```

Arguments

- object A bipl5_biplot object
- eigenvectors Integer vector of length 2 giving the PC pair (e.g. c(4, 5))

Value

A new bipl5_biplot with the additional mdsDisplay appended

bipl5

bipl5: Constructing Reactive Calibrated Axes Biplots

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. For more information on biplots, see Gower, J.C., Lubbe, S. and le Roux, N.J. (2011, ISBN: 978-0-470-01255-0)

Details

Package: bipl5
Type: Package
Version: 1.1
Date: 18-12-2023
License: MIT

Author(s)

- Ruan Buys (Maintainer)
- Carel van der Merwe
- Delia Sandilands (contributer)
- Sugnet Lubbe (contributer)

Core Functions

- [init_biplot](#)
- [scale_mds](#)
- [wrap_bipl5](#)
- [extract](#)
- [append_mdsDisplay](#)
- [remove_mdsDisplay](#)
- [plot.bipl5_biplot](#)
- [plot.bipl5_fit](#)

Code Availability

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

colorpal	<i>bipl5 default color scales</i>
----------	-----------------------------------

Description

bipl5 default color scales

Usage

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

Arguments

number Integer - number of distinct colors to return. Ranges from 1 to 16.

Value

Character vector of default colors in bipl5. There are sixteen unique colors defined.

Examples

```
colorpal(number=7)
```

extract	<i>Extract nested components from a bipl5_biplot object</i>
---------	---

Description

Three calling styles are supported:

1. **mdsDisplay subset:** `extract(bp, mdsDisplay_12)` — returns a new `bipl5_biplot` containing only that `mdsDisplay` (plottable).
2. **Two-level:** `extract(bp, from = mdsDisplay_12, what = sample_coordinates)` — returns the nested data element.
3. **Arbitrary depth:** `extract(bp, mdsDisplay_12$Data$sample_coordinates)` — returns the nested data element.

Usage

```
extract(object, expr, from, what)
```

```
## S3 method for class 'bipl5_biplot'
extract(object, expr, from, what)
```

Arguments

object	A biplot object
expr	An unquoted mdsDisplay name (e.g. mdsDisplay_12) or a path expression using \$ (e.g. mdsDisplay_12\$Data\$sample_coordinates or fit_measures\$CumPred)
from	Unquoted name of the top-level element
what	Unquoted name of the nested element

Details

In addition to the mdsDisplay access patterns above, graph-based fit measures can be extracted directly with calls such as `extract(bp, fit_measures, CumPred)` or `extract(bp, fit_measures$CumPred)`. Supported graph-based fit measures are `CumPred`, `CumAd`, `VarExp`, and `Scree`. These calls return a `biplot_fit` object that can be passed to `plot()` to obtain a static **ggplot2** version of the corresponding fit graph.

Value

A `biplot` (mdsDisplay subset), a `biplot_fit` object for graph-based fit measures, or the requested sub-element.

Examples

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

only_12 <- extract(bp, mdsDisplay_12)
data_obj <- extract(bp, from = mdsDisplay_12, what = Data)
coords <- extract(bp, mdsDisplay_12$Data$sample_coordinates)

fit_plot <- extract(bp, fit_measures, CumPred)
plot(fit_plot)
```

format_samples

Format sample aesthetics on a biplot

Description

`format_samples()` rebuilds the sample-trace block inside each `mdsDisplay` so that observations are grouped by by and rendered with one trace per visual class. This means the visible trace structure, legend labels, and stored sample-format metadata all stay aligned.

Usage

```
format_samples(
  x,
  stratify = c("col", "symbol"),
  by = NULL,
  col = NULL,
  pch = NULL
)
```

Arguments

x	A <code>biplot5</code> object.
stratify	Which aesthetic to change: "col" for marker colour or "symbol" for marker symbol.
by	Optional grouping variable for the sample traces. This can be: • a bare column name stored in the dataset supplied to <code>init_biplot()</code> • a single character column name stored in the object, or • a vector/factor of length n, one value per observation. When NULL, the current sample grouping in <code>x\$meta\$group</code> is reused.
col	Optional vector of colours. When <code>stratify = "col"</code> , this must have one value per visual class defined by <code>by</code> . If omitted, a default palette is used.
pch	Optional vector of plotting symbols. When <code>stratify = "symbol"</code> , this must have one value per visual class defined by <code>by</code> . Numeric base-R pch codes are converted internally to plotly symbols; character plotly symbol names are also accepted.

Details

The function is intended for sample formatting only. It does not refit the underlying ordination model. In particular, for CVA biplots the fitted CVA classes are preserved and only the sample traces are reformatted.

A first call to `format_samples()` creates one sample legend section for the requested aesthetic. For example, `format_samples(stratify = "col", by = Species)` will colour the observations by Species and create a legend section headed Species with one entry per class.

A second call can be used to add a second, independent sample stratification. If the second call uses the same grouping variable as the first call, both aesthetics are applied to the same set of classes and the legend remains unified. If the second call uses a different grouping variable, `format_samples()` creates a second legend section and internally splits the observation layer into all observed combinations of the two grouping variables.

For example, the sequence

```
init_biplot(iris2) |> scale_mds("pca") |> format_samples(stratify = "col", by = Species)
|> format_samples(stratify = "symbol", by = Band)
```

will produce two sample legend sections:

- Species for the colour grouping

- Band for the symbol grouping

The visible observation traces are then split by Species $\times$ Band, but these combination traces are hidden from the legend. Instead, `format_samples()` inserts legend-only sample entries so the legend remains easy to read.

If translated axes are available in the `mdsDisplay`, a colour stratification also rebuilds the kernel-density traces on the translated axes so that those densities reflect the colour classes. Symbol-only stratification does not change the translated-axis densities. This means:

- `format_samples(stratify = "col", ...)` recalculates translated-axis densities by the colour grouping
- `format_samples(stratify = "symbol", ...)` leaves the existing translated densities unchanged

The legend toggles operate across the full dual stratification:

- clicking a colour legend entry hides or shows all observations belonging to that colour class, across every symbol class
- clicking a symbol legend entry hides or shows all observations belonging to that symbol class, across every colour class

The formatting is applied to every `mdsDisplay` currently stored in the object. If additional displays are later added with `append_mdsDisplay()`, the stored sample-format state is reused so the new displays inherit the same sample legend structure.

`format_samples()` supports two complementary workflows.

Single stratification

A single call to `format_samples()` rebuilds the sample layer so that one trace is created per class in by. This updates the marker appearance, the legend entries, and the stored sample-format metadata consistently.

Second stratification

A second call to `format_samples()` can be used to add a second sample aesthetic. This is most useful when colour and plotting symbol represent different variables.

If the second call uses the same grouping structure as the first, the result is still one legend section with one entry per class, but each class now carries both a colour and a plotting symbol.

If the second call uses a different grouping structure, the object stores two independent sample legend sections. Internally, the observation layer is rebuilt as one hidden trace per observed combination of the two grouping variables. The visible legend then shows one section for each stratifying variable.

Translated-axis densities

When translated axes are present, the kernel-density traces on those axes are tied to the current colour grouping. Applying `format_samples(stratify = "col", ...)` rebuilds the translated-axis density traces so they match the colour classes. Applying `format_samples(stratify = "symbol", ...)` does not rebuild those densities.

So:

- a first colour stratification updates both the sample layer and the translated-axis densities
- a later symbol stratification leaves those densities as they are

- if a symbol stratification is applied first and a colour stratification is added later, the translated-axis densities are rebuilt when the colour stratification is added

Legend click behaviour

When two different stratifications are active, the legend entries behave like filters:

- clicking a class in the first legend section toggles all observations in that class, regardless of their membership in the second stratification
- clicking a class in the second legend section toggles all observations in that class, regardless of their membership in the first stratification

So if colours represent Species and symbols represent Band, clicking *setosa* hides all *setosa* observations, while clicking *class1* hides all *class1* observations across every species.

Non-standard evaluation

If *by* is supplied as a bare column name, `format_samples()` looks for that column in the dataset stored by `init_biplot()`. If *by* is supplied as a character string, it is interpreted as the name of a stored column. If *by* is supplied as a vector, it must have one value per observation; in that case the legend title defaults to "Data" because there is no stored column name to display.

CVA note

`format_samples()` does not change the fitted CVA model. It only reformats the sample traces. The grouping used to fit the CVA model should therefore be specified in `scale_mds()`, not in `format_samples()`.

Value

A modified `biplot`.

Examples

```
bp <- init_biplot(iris) |>
  scale_mds(type = "pca", eigenvectors = c(1, 2))

bp_species <- format_samples(
  bp,
  stratify = "col",
  by = Species,
  col = c("tomato", "steelblue", "darkgreen")
)

sample_idx <- vapply(
  bp_species$mdsDisplay_12$mdsDisplay$trace_data,
  function(tr) "data" %in% unlist(tr$meta),
  logical(1)
)

vapply(
  bp_species$mdsDisplay_12$mdsDisplay$trace_data[sample_idx],
  `[`,
  character(1),
  "name"
)
```

```

bp_symbol <- format_samples(
  bp,
  stratify = "symbol",
  by = Species,
  pch = c(16, 17, 15)
)

iris2 <- iris
iris2$Band <- factor(
  rep(c("class1", "class2", "class3", "class4"), length.out = nrow(iris2))
)

bp_dual <- init_biplot(iris2) |>
  scale_mds(type = "pca", eigenvectors = c(1, 2)) |>
  format_samples(
    stratify = "col",
    by = Species,
    col = c("tomato", "steelblue", "darkgreen")
  ) |>
  format_samples(
    stratify = "symbol",
    by = Band,
    pch = c(12, 13, 14, 15)
  )

# When plotted, the legend now has one section for Species and one for Band.
# Clicking a Species entry hides that species across all Band classes.
# Clicking a Band entry hides that Band class across all Species classes.
if (interactive()) {
  plot(bp_dual)
}

bp_species_13 <- append_mdsDisplay(bp_species, c(1, 3))

```

geom_calibrated_axis *Calibrated biplot axes for ggplot2*

Description

Draws calibrated (Gower) biplot axes: a line spanning the plotting window, orthogonal tick marks at calibrated marker positions, marker values rotated to the angle of the axis, and the variable name written at the window border.

Usage

```

geom_calibrated_axis(
  mapping = NULL,
  data = NULL,
  stat = "identity",

```

```

position = "identity",
axis_type = c("line", "curve"),
axis_line = TRUE,
axis_labels = TRUE,
axis_ticks = TRUE,
axis_text = TRUE,
tick_extend = TRUE,
tick_length = 0.025,
text_dodge = 0.03,
label_dodge = 0.03,
label_placement = c("positive", "negative", "peripheral"),
parse = FALSE,
check_overlap = FALSE,
na.rm = FALSE,
show.legend = NA,
inherit.aes = TRUE,
...
)

```

Arguments

mapping	Set of aesthetic mappings created by aes .
data	The data to be displayed in this layer.
stat	The statistical transformation to use on the data.
position	Position adjustment.
axis_type	Either "line" for straight axes or "curve" for the polyline (spline) axes of a principal coordinate analysis biplot.
axis_line, axis_labels, axis_ticks, axis_text	Logical; whether to draw the axis line, the variable name, the tick marks, and the marker values.
tick_extend	Logical; if TRUE (the default) markers are regenerated from the recovered calibration so that they cover the whole visible axis, preserving the marker spacing and phase chosen by biplotEZ . If FALSE only the supplied markers are drawn. Ignored when axis_type = "curve".
tick_length	Length of the tick marks, as a proportion of the smaller of the panel width and height.
text_dodge	Orthogonal distance of the marker values from the axis, as a proportion of the smaller of the panel width and height.
label_dodge	Orthogonal distance of the variable name from the axis, as a proportion of the smaller of the panel width and height.
label_placement	One of "positive" (the default; the increasing end of the axis), "negative", or "peripheral" (the end farther from the origin).
parse, check_overlap	Passed to the underlying text grobs.

na.rm Logical; silently remove missing values.
 show.legend, inherit.aes Standard **ggplot2** layer arguments.
 ... Additional arguments passed to [layer](#).

Details

The layer is designed to consume the calibrated axis coordinates returned by `biplotEZ::axes_coordinates()`, which [wrap_bipl5_gg](#) maps onto the aesthetics documented below. Because the marker positions are supplied rather than derived, the layer honours whatever calibration **biplotEZ** produced, including axes translated away from the origin and the curved spline axes used by principal coordinate analysis biplots.

Value

A **ggplot2** layer.

Aesthetics

`geom_calibrated_axis()` understands the following aesthetics (required aesthetics are in bold):

- **x**, **y** — calibrated marker positions in display space. These do *not* contribute to the plotting window, so axes never inflate the panel range.
- **label** — the marker value printed next to each tick.
- **axis_label** — the variable name written at the window border.
- **angle** — direction of increasing variable value, in radians.
- **tick** — logical flag marking which rows carry a tick mark; used by curved axes, where most rows are only curve vertices.
- **value** — the variable value at each marker, used to identify the increasing end of a curved axis.
- **cal_x0**, **cal_y0**, **cal_dxdv**, **cal_dydv**, **cal_vref**, **cal_vstep** — the affine calibration recovered from `biplotEZ::axes_coordinates()`, used when `tick_extend = TRUE`.
- **off_x**, **off_y** — the foot of the perpendicular from the origin to a translated axis.
- **axis_colour**, **axis_linewidth**, **axis_linetype**, **axis_alpha**
- **tick_colour**, **tick_linewidth**
- **text_colour**, **text_size**
- **label_colour**, **label_size**
- **family**, **fontface**, **group**

Element aesthetics left at NA inherit from the axis line, mirroring the `sync()` fallbacks of **gggda**.

References

Gower, J. C. and Hand, D. J. (1996). *Biplots*. London: Chapman and Hall.
 Gower, J. C., Lubbe, S. and le Roux, N. J. (2011). *Understanding Biplots*. Chichester: Wiley.
 Brunson, J. C. (2025). *gggda: A 'ggplot2' Extension for Geometric Data Analysis*. R package version 0.2.0.

See Also

[wrap_bipl5_gg](#), which assembles a complete biplot around this layer.

Examples

```
## Not run:
library(ggplot2)
library(biplotEZ)

bp <- biplot(iris[, 1:4], scale = TRUE) |>
  PCA(group.aes = iris[, 5]) |>
  axes()

axis_df <- bipl5::calibrated_axis_frame(bp)$data

ggplot() +
  geom_point(
    data = data.frame(x = bp$Z[, 1], y = bp$Z[, 2]),
    mapping = aes(x = x, y = y)
  ) +
  geom_calibrated_axis(
    data = axis_df,
    mapping = aes(
      x = x, y = y, group = group, angle = angle,
      label = label, axis_label = axis_label
    )
  ) +
  coord_equal()

## End(Not run)
```

init_biplot

Create a bipl5 specification object

Description

`init_biplot()` stores the raw data and preprocessing options needed to construct a biplot later with [scale_mds\(\)](#). It does not perform any ordination itself. When data is a data frame containing both numeric and non-numeric columns, only the numeric columns are used for the biplot calculation, while the full data frame is retained for later formatting steps such as [format_samples\(\)](#).

Usage

```
init_biplot(data, center = TRUE, scale = FALSE)
```

Arguments

data	A matrix or data frame. If a data frame contains non-numeric columns, they are stored but excluded from the ordination input.
center	Logical; should numeric variables be centered before analysis?
scale	Logical; should numeric variables be scaled before analysis?

Value

An object of class `bip15_spec`.

overlay_fit	<i>Store a default fit-measure display mode on a biplot</i>
-------------	---

Description

`overlay_fit()` is a convenience helper for pipelines. It does not refit the underlying ordination; it only stores whether fit measures should default to the right-hand panel or an overlay view when `plot()` is called.

Usage

```
overlay_fit(x, overlay = TRUE)
```

Arguments

x	A <code>bip15_biplot</code> object with fit measures.
overlay	Logical scalar. TRUE stores "overlay"; FALSE stores "panel".

Details

A later call to `plot(x, fit_display = ...)` always takes precedence over the stored default.

Value

A modified `bip15_biplot`.

plot.bipl5_biplot	<i>Plot a bipl5_biplot object</i>
-------------------	-----------------------------------

Description

Initialises a plotly graph, populates it with the first available mdsDisplay traces and annotations, then attaches the remaining mdsDisplays and fit measures to the JavaScript event handler.

Usage

```
## S3 method for class 'bipl5_biplot'
plot(x, y = NULL, fit_display = c("inherit", "panel", "overlay"), ...)
```

Arguments

x	A bipl5_biplot object
y	Ignored (for S3 consistency)
fit_display	How fit measures should be shown for biplots that support them: inherit the object's stored preference, render them in the right-hand panel, or render them as an overlay over the full plot width.
...	Additional arguments (ignored)

Value

A plotly htmlwidget

plot.bipl5_fit	<i>Plot a single extracted fit graph as a ggplot</i>
----------------	--

Description

Reconstructs one of the PCA fit graphs from its stored plotly traces. The fit type is inferred from the trace metadata and trace types, then translated into a **ggplot2** chart with matching title, legend titles, and axes.

Usage

```
## S3 method for class 'bipl5_fit'
plot(x, y = NULL, ...)
```

Arguments

x	A bipl5_fit object, typically returned by extract(bp, fit_measures, CumPred) or a similar fit graph extraction.
y	Ignored (for S3 consistency)
...	Additional arguments (ignored)

Details

Supported fit graphs are cumulative predictivity (CumPred), cumulative adequacy (CumAd), variance explained (VarExp), and the scree plot (Scree). The summary-table fit objects are not handled by this plotting method.

Value

A ggplot object.

Examples

```
bp <- biplotEZ::biplot(iris[, 1:4]) |>
  biplotEZ::PCA() |>
  wrap_bipl5()

fit_plot <- extract(bp, fit_measures, Scree)
plot(fit_plot)
```

print.bipl5_biplot	<i>Print a bipl5_biplot object as a tree diagram</i>
--------------------	--

Description

Print a bipl5_biplot object as a tree diagram

Usage

```
## S3 method for class 'bipl5_biplot'
print(x, ...)
```

Arguments

x	A bipl5_biplot object
...	Additional arguments (ignored)

Value

Invisibly returns x

print.bipl5_data	<i>Print a bipl5_data object</i>
------------------	----------------------------------

Description

Print a bipl5_data object

Usage

```
## S3 method for class 'bipl5_data'  
print(x, ...)
```

Arguments

x	A bipl5_data object
...	Additional arguments (ignored)

Value

Invisibly returns x

print.bipl5_fitmeasures	<i>Print a bipl5_fitmeasures object</i>
-------------------------	---

Description

Print a bipl5_fitmeasures object

Usage

```
## S3 method for class 'bipl5_fitmeasures'  
print(x, ...)
```

Arguments

x	A bipl5_fitmeasures object
...	Additional arguments (ignored)

Value

Invisibly returns x

```
print.bipl5_mdsDisplay
```

Print a bipl5_mdsDisplay object

Description

Print a bipl5_mdsDisplay object

Usage

```
## S3 method for class 'bipl5_mdsDisplay'
print(x, ...)
```

Arguments

x	A bipl5_mdsDisplay object
...	Additional arguments (ignored)

Value

Invisibly returns x

```
remove_mdsDisplay
```

Remove a mdsDisplay from a bipl5_biplot object

Description

Returns a new bipl5_biplot with the specified mdsDisplay (and its corresponding fit table) removed. At least one mdsDisplay must remain.

Usage

```
remove_mdsDisplay(object, mdsDisplay)

## S3 method for class 'bipl5_biplot'
remove_mdsDisplay(object, mdsDisplay)
```

Arguments

object	A bipl5_biplot object
mdsDisplay	Unquoted name of the mdsDisplay to remove (e.g. mdsDisplay_13)

Value

A new bipl5_biplot without the removed mdsDisplay

scale_mds

*Scale a biplot specification into a biplot5_biplot***Description**

scale_mds() turns a biplot5_spec created by `init_biplot()` into a fully formed biplot5_biplot by dispatching to one of the underlying `biplotEZ::PCA()`, `biplotEZ::CVA()`, `biplotEZ::PCO()`, or `biplotEZ::regress()` methods and then compiling only the requested mdsDisplay. Any additional displays can be added later with `append_mdsDisplay()`.

Usage

```
scale_mds(x, type = c("pca", "cva", "pco", "regress"), ...)
```

```
## S3 method for class 'biplot5_spec'
```

```
scale_mds(x, type = c("pca", "cva", "pco", "regress"), ...)
```

Arguments

x	A biplot5_spec created by <code>init_biplot()</code> .
type	The biplot method to construct. One of "pca", "cva", "pco", "regress", or "regression".
...	Additional named arguments for the chosen method.

Details

The type argument chooses the underlying biplot method. Additional arguments are method-specific and should be supplied via

Supported aliases in ...:

- Common: `classes`, `group_aes` / `group.aes`, `title` / `Title`
- PCA: `dimensions` / `dim.biplot`, `eigenvectors` / `e.vects`, `show_class_means` / `show.class.means` / `show_group_means` / `show.group.means`, `correlation_biplot` / `correlation.biplot`
- CVA: `classes`, `dimensions` / `dim.biplot`, `eigenvectors` / `e.vects`, `weighted_cva` / `weightedCVA`, `show_class_means` / `show.class.means` / `show_group_means` / `show.group.means`, `low_dim` / `low.dim`
- PCO: `Dmat` / `dist_mat`, `dist_func` / `dist.func`, `dist_func_cat` / `dist.func.cat`, `dimensions` / `dim.biplot`, `eigenvectors` / `e.vects`, `show_class_means` / `show.class.means` / `show_group_means` / `show.group.means`, `axes`
- regress: `Z` / `z`, `show_group_means` / `show.group.means` / `show_class_means` / `show.class.means`, `axes`

For type = "pco", any remaining named arguments in ... are forwarded to the chosen distance function.

Value

A fully formed biplot5_biplot.

score_axes	<i>Annotate a biplot with Alves direct-reading errors</i>
------------	---

Description

score_axes() augments the hover information of every mdsDisplay in a biplot5_biplot with the direct-reading diagnostic of Alves (2012). For each observation i and variable j the direct-reading error is

Usage

```
score_axes(x, digits = 2, ...)

## S3 method for class 'biplot5_biplot'
score_axes(x, digits = 2, ...)
```

Arguments

x	A biplot5_biplot produced by <code>scale_mds()</code> .
digits	Number of decimal places used when displaying the reading error percentage. Defaults to 2.
...	Currently unused.

Details

$$\delta_{ij} = \frac{|x_{ij} - \hat{x}_{ij}|}{s_j},$$

where x_{ij} is the actual value, $\hat{x}_{ij}$ is the value read directly off the calibrated axis for variable j (i.e. the orthogonal projection of the sample point onto the displayed axis), and s_j is the scaling constant used when the biplot was drawn. The scaling constant is 1 when the data were not scaled and the column standard deviation s_j when `scale = TRUE` was passed to `init_biplot()`. The quantity is reported in percentage form ($100 \delta_{ij}$) as an extra Error column in the hover table shown when a data point is hovered over.

The diagnostic is computed separately for each mdsDisplay present in the object, because the read-off value $\hat{x}_{ij}$ depends on the dimension pair being displayed. The methodology applies uniformly to PCA, PCO and CVA biplots, which all use calibrated linear axes. Spline (non-linear) PCO axes do not admit a single calibrated reading and are left unchanged.

This step is deliberately *not* performed by default: it is only applied when `score_axes()` is inserted into the pipeline, e.g.

```
init_biplot(data) |> scale_mds() |> score_axes() |> plot()
```

Value

A bipl5_biplot whose observation hover tables carry an additional Error column. The object remains fully plottable.

References

Alves, M. R. (2012). Evaluation of the predictive power of biplot axes to automate the construction and layout of biplots based on the accuracy of direct readings from common outputs of multivariate analyses: application to principal component analysis. *Journal of Chemometrics*, 26(5), 180-190.

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

Description

Retrieve all valid plotting symbols for the plotly library

Usage

```
Symbol_List()
```

Value

A vector of all the valid plotting symbols used in the [plot_ly](#) library.

Examples

```
Symbol_List()
```

wrap_bipl5	<i>Convert a biplotEZ object to a bipl5_biplot</i>
------------	--

Description

Convert a biplotEZ object to a bipl5_biplot

Usage

```
wrap_bipl5(x)
```

Arguments

x A biplotEZ biplot object

Value

An object of class bipl5_biplot

wrap_bipl5.CVA	<i>Construct a bipl5_biplot from a CVA biplot</i>
----------------	---

Description

Builds mdsDisplays for the user's CV pair and available supplementary pairs, along with a drop-down menu. Fit measures are not yet computed for CVA biplots and will be NULL. Plotting is deferred to [plot.bipl5_biplot](#).

Usage

```
## S3 method for class 'CVA'
wrap_bipl5(x)
```

Arguments

x An object of class biplot from the biplotEZ package with CVA method applied.

Value

An object of class c("bipl5_biplot", "cva")

Examples

```
## Not run:
library(biplotEZ)
bp <- biplot(iris[, 1:4]) |> CVA(classes = iris[, 5]) |> wrap_bipl5()
bp
plot(bp)

## End(Not run)
```

wrap_bipl5.PCA	<i>Construct a bipl5_biplot from a PCA biplot</i>
----------------	---

Description

Builds the mdsDisplay(s) used for a principal component analysis (PCA) biplot and documents the associated PCA-biplot fit, predictivity and direct-reading measures. In contrast to a regression biplot, the low-dimensional sample map is obtained internally from the singular value decomposition of the processed data matrix. If the wrapped object stores more than one principal-component pair as separate mdsDisplays, the same formulas below apply to each mdsDisplay separately, with the active mdsDisplay determined by the displayed pair of principal components.

Usage

```
## S3 method for class 'PCA'
wrap_biplot(x)
```

Arguments

x An object of class biplot from the **biplotEZ** package with PCA() method applied.

Details

For the PCA biplot handled by this method, let $\mathbf{X} \in \mathbb{R}^{n \times p}$ denote the *processed* data matrix stored in the input biplot object after centring and any optional scaling performed by biplot(). Thus $\mathbf{X}$ is the matrix on which PCA is actually carried out. Write the singular value decomposition as

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^\top,$$

where $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$, $\mathbf{V}^\top\mathbf{V} = \mathbf{I}$, and $\mathbf{D} = \text{diag}(d_1, \dots, d_q)$ with $q = \text{rank}(\mathbf{X})$ and $d_1 \geq \dots \geq d_q \geq 0$. The columns of $\mathbf{V}$ are the principal directions, and the corresponding principal component score vectors are $\mathbf{z}_t = d_t\mathbf{u}_t = \mathbf{X}\mathbf{v}_t$, $t = 1, \dots, q$. This is the standard PCA biplot construction underlying Gabriel's original formulation and subsequent calibrated-axis developments (Gabriel, 1971; Gower and Hand, 1996; Gower, Lubbe and le Roux, 2011; Greenacre, 2010).

Suppose the user has selected two principal components $a < b$. Let $\mathbf{J}_{ab}$ denote the diagonal $q \times q$ selector matrix with ones in positions a and b and zeros elsewhere. Then the two-dimensional PCA fitted matrix is

$$\hat{\mathbf{X}}_{ab} = \mathbf{U}\mathbf{D}\mathbf{J}_{ab}\mathbf{V}^\top = \mathbf{X}\mathbf{V}\mathbf{J}_{ab}\mathbf{V}^\top.$$

Equivalently, if $\mathbf{U}_{ab}$ and $\mathbf{V}_{ab}$ denote the submatrices containing columns a and b , and $\mathbf{D}_{ab} = \text{diag}(d_a, d_b)$, then

$$\hat{\mathbf{X}}_{ab} = \mathbf{U}_{ab}\mathbf{D}_{ab}\mathbf{V}_{ab}^\top = d_a\mathbf{u}_a\mathbf{v}_a^\top + d_b\mathbf{u}_b\mathbf{v}_b^\top.$$

When $(a, b) = (1, 2)$, this is the best rank-2 approximation to $\mathbf{X}$ in Frobenius norm by the Eckart–Young theorem. For any other selected pair, the same formula gives the orthogonal projection of $\mathbf{X}$ onto the chosen two-dimensional principal-component subspace, but it is not generally the globally optimal rank-2 approximation (Eckart and Young, 1936; Gabriel, 1971; Greenacre, 2010).

The calibrated-axis PCA biplot may be written in the general form

$$\hat{\mathbf{X}}_{ab} = \mathbf{Z}_{ab}\mathbf{H}_{ab}^\top,$$

where the exact factorization depends on the type of PCA biplot being displayed.

For the ordinary PCA biplot, which prioritizes the Euclidean geometry of the sample points, take

$$\mathbf{Z}_{ab} = \mathbf{U}_{ab}\mathbf{D}_{ab}, \quad \mathbf{H}_{ab} = \mathbf{V}_{ab}.$$

Thus the displayed sample coordinates are the selected principal component scores, and the fitted matrix is

$$\hat{\mathbf{X}}_{ab} = \mathbf{Z}_{ab}\mathbf{H}_{ab}^\top.$$

If $\mathbf{h}_{(j)}$ denotes the j th row of $\mathbf{H}_{ab}$ written as a column vector in $\mathbb{R}^2$, then for sample i

$$\hat{x}_{ij} = \mathbf{z}_i^\top \mathbf{h}_{(j)}.$$

Hence the calibrated axis for variable j has direction $\mathbf{h}_{(j)}$, and the point on that axis corresponding to marker value μ is

$$\mathbf{p}_{\mu j} = \frac{\mu}{\mathbf{h}_{(j)}^\top \mathbf{h}_{(j)}} \mathbf{h}_{(j)}.$$

This is the standard calibrated-axis formula used to place tick marks and to recover predicted values from projections onto the displayed axis (Gower and Hand, 1996; Gower, Lubbe and le Roux, 2011).

A second important special case is the *correlation biplot*, obtained when the processed matrix $\mathbf{X}$ is standardized and the display is chosen so that correlations between variables are approximated by the cosines of the angles between the displayed variable directions. In that case one may equivalently factorize

$$\hat{\mathbf{X}}_{ab} = \mathbf{U}_{ab} (\mathbf{V}_{ab} \mathbf{D}_{ab})^\top.$$

Hence the displayed sample coordinates are $\mathbf{U}_{ab}$ and the variable directions are the rows of $\mathbf{V}_{ab} \mathbf{D}_{ab}$. If $\mathbf{c}_{(j)}$ denotes the j th such row written as a column vector, then

$$\hat{x}_{ij} = \mathbf{u}_{i,ab}^\top \mathbf{c}_{(j)}.$$

In this standardized setting the coordinates $\mathbf{c}_{(j)}$ are proportional to the correlations of variable j with the selected principal components, and the geometry of the displayed variable directions is therefore tuned to the correlation structure rather than to the raw score geometry of the samples. This is the sense in which `correlation.biplot = TRUE` preserves variable-correlation information in the display (Gabriel, 1971; Greenacre, 2010; **biplotEZ** manual and vignette).

In either factorization, all predicted values in $\hat{\mathbf{X}}_{ab}$ are on the same centred/scaled scale as the stored matrix $\mathbf{X}$. If required, predictions can be back-transformed to the original variable scale using the means and standard deviations stored in the input biplot object.

A fundamental feature of the PCA biplot is that both *sample-side* and *variable-side* orthogonal decompositions hold. Writing $\mathbf{E}_{ab} = \mathbf{X} - \hat{\mathbf{X}}_{ab}$, one has

$$\mathbf{X} \mathbf{X}^\top = \hat{\mathbf{X}}_{ab} \hat{\mathbf{X}}_{ab}^\top + \mathbf{E}_{ab} \mathbf{E}_{ab}^\top,$$

and

$$\mathbf{X}^\top \mathbf{X} = \hat{\mathbf{X}}_{ab}^\top \hat{\mathbf{X}}_{ab} + \mathbf{E}_{ab}^\top \mathbf{E}_{ab}.$$

The first is the *Type A orthogonality*, which justifies sample-side measures of fit. The second is the *Type B orthogonality*, which justifies variable-side measures of fit. For PCA both orthogonality relations hold simultaneously because $\hat{\mathbf{X}}_{ab}$ is obtained from an orthogonal principal-component projection (Gabriel, 1971; Gower, Lubbe and le Roux, 2011; Gardner-Lubbe, le Roux and Gower, 2008).

Let $\mathbf{x}_{i\cdot}^\top$ denote row i of $\mathbf{X}$ and $\hat{\mathbf{x}}_{i\cdot}^\top$ the corresponding row of $\hat{\mathbf{X}}_{ab}$. The *sample predictivity* of sample i is then

$$\psi_i = \frac{\|\hat{\mathbf{x}}_{i\cdot}\|^2}{\|\mathbf{x}_{i\cdot}\|^2} = 1 - \frac{\|\mathbf{x}_{i\cdot} - \hat{\mathbf{x}}_{i\cdot}\|^2}{\|\mathbf{x}_{i\cdot}\|^2}, \quad i = 1, \dots, n.$$

Thus ψ_i is the proportion of the sum of squares of sample i reproduced by the chosen two-dimensional PCA display. Because of Type A orthogonality, $0 \leq \psi_i \leq 1$. Samples with ψ_i near one lie close to the displayed PCA plane, whereas samples with ψ_i near zero lie largely orthogonal to it. This is the sample-side fit measure used in the PCA biplot literature and in **biplotEZ** (Gardner-Lubbe, le Roux and Gower, 2008; **biplotEZ** vignette).

Let $\mathbf{x}_{(j)}$ denote column j of $\mathbf{X}$ and $\hat{\mathbf{x}}_{(j)}$ the corresponding column of $\hat{\mathbf{X}}_{ab}$. The *axis predictivity* of variable j is

$$\phi_j = \frac{\|\hat{\mathbf{x}}_{(j)}\|^2}{\|\mathbf{x}_{(j)}\|^2} = 1 - \frac{\|\mathbf{x}_{(j)} - \hat{\mathbf{x}}_{(j)}\|^2}{\|\mathbf{x}_{(j)}\|^2}, \quad j = 1, \dots, p.$$

Thus ϕ_j is the proportion of the sum of squares of variable j reproduced by the chosen PCA plane. Because of Type B orthogonality, $0 \leq \phi_j \leq 1$. In a calibrated-axis display, ϕ_j is the natural sum-of-squares measure of how well the axis for variable j reproduces the underlying processed values. This is the quantity reported in **biplotEZ** as “axis predictivity” (Gardner-Lubbe, le Roux and Gower, 2008; **biplotEZ** vignette; Greenacre, 2010).

The overall quality of the displayed PCA subspace is

$$R_{\text{disp},ab}^2 = \frac{\|\hat{\mathbf{X}}_{ab}\|_F^2}{\|\mathbf{X}\|_F^2} = 1 - \frac{\|\mathbf{X} - \hat{\mathbf{X}}_{ab}\|_F^2}{\|\mathbf{X}\|_F^2}.$$

Since $\|\hat{\mathbf{X}}_{ab}\|_F^2 = d_a^2 + d_b^2$, this may also be written as

$$R_{\text{disp},ab}^2 = \frac{d_a^2 + d_b^2}{\sum_{t=1}^q d_t^2}.$$

In particular, when $(a, b) = (1, 2)$, this is the familiar proportion of total sum of squares explained by the first two principal components. More generally, it is the quality of the specific displayed pair chosen by the user, matching the **biplotEZ** quality measure (Gabriel, 1971; Greenacre, 2010; **biplotEZ** vignette).

Because both Type A and Type B orthogonality hold, the overall quality can be expressed as a weighted average on either the sample side or the variable side. On the variable side,

$$R_{\text{disp},ab}^2 = \sum_{j=1}^p w_j \phi_j,$$

where

$$w_j = \frac{\|\mathbf{x}_{(j)}\|^2}{\|\mathbf{X}\|_F^2}, \quad \sum_{j=1}^p w_j = 1.$$

Hence variables with larger sums of squares contribute more to the overall display quality. If the original call to `biplot()` used `scale = TRUE`, so that all processed variables have equal sums of squares, then

$$R_{\text{disp},ab}^2 = \frac{1}{p} \sum_{j=1}^p \phi_j.$$

Thus, for a standardized PCA biplot, the overall quality is the simple average of the individual axis predictivities. This is the weighted-average interpretation requested by the present wrapper.

Similarly, on the sample side,

$$R_{\text{disp},ab}^2 = \sum_{i=1}^n m_i \psi_i,$$

where

$$m_i = \frac{\|\mathbf{x}_i\|^2}{\|\mathbf{X}\|_F^2}, \quad \sum_{i=1}^n m_i = 1.$$

Hence the same overall display quality may be read as a weighted average of sample predictivities or as a weighted average of axis predictivities (Gardner-Lubbe, le Roux and Gower, 2008).

Unlike the regression-biplot case, no ordered orthogonalization is required to decompose the quality of a PCA display into separate contributions from the two displayed dimensions, because principal components are already mutually orthogonal. Indeed,

$$\hat{\mathbf{X}}_{ab} = d_a \mathbf{u}_a \mathbf{v}_a^\top + d_b \mathbf{u}_b \mathbf{v}_b^\top,$$

and these two rank-1 parts are orthogonal in Frobenius inner product. Hence

$$R_{\text{disp},ab}^2 = R_a^2 + R_b^2,$$

where

$$R_a^2 = \frac{d_a^2}{\|\mathbf{X}\|_F^2}, \quad R_b^2 = \frac{d_b^2}{\|\mathbf{X}\|_F^2}.$$

Thus the contribution of each displayed principal component is obtained directly from its singular value.

The same orthogonal decomposition yields a per-component breakdown of each axis predictivity. Since

$$\hat{\mathbf{x}}_{(j)} = d_a v_{ja} \mathbf{u}_a + d_b v_{jb} \mathbf{u}_b,$$

with orthogonal components $\mathbf{u}_a$ and $\mathbf{u}_b$, one has

$$\phi_j = \phi_{ja} + \phi_{jb},$$

where

$$\phi_{ja} = \frac{d_a^2 v_{ja}^2}{\|\mathbf{x}_{(j)}\|^2}, \quad \phi_{jb} = \frac{d_b^2 v_{jb}^2}{\|\mathbf{x}_{(j)}\|^2}.$$

Hence the predictivity of axis j can be decomposed exactly into the separate contributions of the two displayed principal components. This is the PCA analogue of the dimension-wise decomposition used elsewhere in the biplot literature, but here it is especially simple because the components are orthogonal from the outset. In particular, if the same variable is well aligned with one selected principal direction but not the other, this will be visible in the separate values ϕ_{ja} and ϕ_{jb} .

Likewise, each sample predictivity decomposes as

$$\psi_i = \psi_{ia} + \psi_{ib},$$

where

$$\psi_{ia} = \frac{d_a^2 u_{ia}^2}{\|\mathbf{x}_i\|^2}, \quad \psi_{ib} = \frac{d_b^2 u_{ib}^2}{\|\mathbf{x}_i\|^2}.$$

Thus the contribution of each displayed principal component may be read not only globally through R_a^2 and R_b^2 , but also locally through the sample-wise contributions ψ_{ia} and ψ_{ib} and the axis-wise contributions ϕ_{ja} and ϕ_{jb} (Gardner-Lubbe, le Roux and Gower, 2008).

In addition to the sum-of-squares fit measures above, this method may also report direct-reading diagnostics in the sense of Alves (2012). These quantities serve a different purpose from the predictivities ϕ_j and ψ_i . The predictivities measure how much of the variation in $\mathbf{X}$ is reproduced by the selected PCA plane in a sum-of-squares sense. By contrast, the Alves diagnostics measure how accurately values can be read directly from the displayed calibrated axes in the current two-dimensional map. This distinction is central in the predictive biplot literature (Alves, 2012).

Let $\mathbf{g}_{(j)} \in \mathbb{R}^2$ denote the displayed direction of variable axis j under the active PCA factorization. Thus $\mathbf{g}_{(j)} = \mathbf{h}_{(j)}$ for the ordinary PCA biplot and $\mathbf{g}_{(j)} = \mathbf{c}_{(j)}$ for the correlation biplot. Let $\mathbf{z}_i \in \mathbb{R}^2$ denote the corresponding displayed sample coordinate of sample i . Then the value read from the graph on axis j for sample i is

$$\hat{x}_{ij} = \mathbf{z}_i^\top \mathbf{g}_{(j)},$$

and the point on the calibrated axis corresponding to that reading is

$$\mathbf{p}_{ij} = \frac{\hat{x}_{ij}}{\mathbf{g}_{(j)}^\top \mathbf{g}_{(j)}} \mathbf{g}_{(j)}.$$

Thus the direct reading from the displayed PCA axis coincides exactly with the fitted value from the active two-dimensional PCA approximation.

Let s_j denote the standard deviation used to standardize variable j . When the processed matrix $\mathbf{X}$ is already standardized, $s_j = 1$. The pointwise direct-reading error for sample i on axis j is then

$$\delta_{ij} = \frac{|x_{ij} - \hat{x}_{ij}|}{s_j}.$$

If $\mathbf{X}$ is already standardized, then $\delta_{ij} = |x_{ij} - \hat{x}_{ij}|$. The corresponding axis-level mean direct-reading error is

$$\bar{\delta}_j = \frac{1}{n} \sum_{i=1}^n \delta_{ij} = \frac{1}{n} \sum_{i=1}^n \frac{|x_{ij} - \hat{x}_{ij}|}{s_j}.$$

This is the two-dimensional PCA-biplot analogue of the mean standard predictive error of Alves (2012).

Let $\tau_{\text{axis}} > 0$ be a user-specified *tolerance parameter* for axis-level direct-reading error. Then the Alves selection rule becomes

$$\text{retain axis } j \iff \bar{\delta}_j \leq \tau_{\text{axis}}.$$

Likewise, for an observation-level tolerance parameter $\tau_{\text{units}} > 0$, sample i is flagged with respect to axis j whenever

$$\delta_{ij} > \tau_{\text{units}}.$$

Hence $\bar{\delta}_j$ may be used for axis selection and δ_{ij} for observation-level checking, exactly as in the predictive PCA-biplot framework of Alves (2012).

The Alves diagnostics and the PCA predictivities are complementary. The quantity ϕ_j answers the question: “How much of variable j ’s sum of squares is reproduced by the selected PCA plane?” The quantity ψ_i answers the corresponding sample-side question: “How much of sample i ’s sum of squares is reproduced by the selected PCA plane?” By contrast, $\bar{\delta}_j$ answers the distinct question: “How accurately can values of variable j be read directly from the displayed calibrated axis?” Consequently, an axis may have high ϕ_j yet still have non-negligible direct-reading error in the current display, while an axis with only moderate ϕ_j may nevertheless admit acceptable direct readings. In this implementation, the ϕ_j - and ψ_i -families are the primary sum-of-squares fit measures, whereas the Alves quantities $\bar{\delta}_j$ and δ_{ij} provide supplementary, display-specific diagnostics.

In the wrapped `biplot5_biplot` object, these formulas drive the hover-time fitted values $\hat{\mathbf{X}}_{ab}$, the calibrated tick markers for each active PCA `mdsDisplay`, the bottom display-quality label, and the axis/sample fit summaries attached to the active two-dimensional principal-component view. If several PC pairs are stored as separate `mdsDisplays`, the same construction applies to each `mdsDisplay` separately.

Value

An object of class `c("bipl5_biplot", "PCA")`

References

- Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1, 211–218.
- Gabriel, K. R. (1971). The biplot graphical display of matrices with application to principal component analysis. *Biometrika*, 58(3), 453–467. doi:[10.1093/biomet/58.3.453](https://doi.org/10.1093/biomet/58.3.453)
- Gower, J. C. and Hand, D. J. (1996). *Biplots*. London: Chapman & Hall.
- Gower, J. C., Lubbe, S. and le Roux, N. J. (2011). *Understanding Biplots*. Chichester: Wiley.
- Greenacre, M. (2010). *Biplots in Practice*. Bilbao: BBVA Foundation.
- Gardner-Lubbe, S., le Roux, N. J. and Gower, J. C. (2008). Measures of fit in principal component and canonical variate analyses. *Journal of Applied Statistics*, 35(9), 947–965. doi:[10.1080/02664760802185399](https://doi.org/10.1080/02664760802185399)
- Lubbe, S., le Roux, N. J., Nienkemper-Swanepoel, J., Ganey, R., Buys, R., Adams, Z.-M. and Manefeldt, P. (2025). *biplotEZ: EZ-to-Use Biplots*. R package version 2.2.
- Alves, M. R. (2012). Evaluation of the predictive power of biplot axes to automate the construction and layout of biplots based on the accuracy of direct readings from common outputs of multivariate analyses:
1. application to principal component analysis. *Journal of Chemometrics*, 26(5), 180–190. doi:[10.1002/cem.2433](https://doi.org/10.1002/cem.2433)

Examples

```
## Not run:
library(biplotEZ)
bp <- biplot(iris[, 1:4], scale = TRUE) |>
  PCA(e.vects = c(1, 2), group.aes = iris[, 5]) |>
  wrap_bipl5()
bp
plot(bp)

bp_cor <- biplot(iris[, 1:4], scale = TRUE) |>
  PCA(
    e.vects = c(1, 2),
    group.aes = iris[, 5],
    correlation.biplot = TRUE
  ) |>
  wrap_bipl5()
plot(bp_cor)

## End(Not run)
```

wrap_bipl5.PCO

Construct a bipl5_biplot from a PCO biplot

Description

Handles two cases depending on the axis type stored in x\$PCOaxes:

Linear axes Built identically to regression biplots via build_one_mdsDisplay(), including translated density axes.

Spline axes Uses a custom mdsDisplay builder (build_spline_mdsDisplay()) that places only sample points, the spline axis curves with tick marks, and a bounding circle. A custom JavaScript handler is attached at plot time.

In both cases there is a single mdsDisplay (mdsDisplay_12), no fit measures, and append_mdsDisplay() / remove_mdsDisplay() are disabled.

Usage

```
## S3 method for class 'PCO'
wrap_bipl5(x)
```

Arguments

x An object of class biplot from the biplotEZ package with PCO() method applied.

Value

An object of class c("bipl5_biplot", "pco")

Examples

```
## Not run:
library(biplotEZ)
bp <- biplot(iris[, 1:4]) |>
  PCO(dist.func = stats::dist) |>
  wrap_bipl5()
bp
plot(bp)

## End(Not run)
```

wrap_bipl5.regress	<i>Construct a bipl5_biplot from a regression biplot</i>
--------------------	--

Description

Builds the single mdsDisplay used for a linear regression biplot and documents the associated regression-biplot fit and predictivity measures. Regression biplots do not use the multi-mdsDisplay fit machinery available for PCA/CVA displays: they have one fixed mdsDisplay (mdsDisplay_12), append_mdsDisplay() and remove_mdsDisplay() are not supported, and the only active toggle button is “Translated Axes”.

Usage

```
## S3 method for class 'regress'
wrap_bipl5(x)
```

Arguments

x	An object of class biplot from the biplotEZ package with regress() method applied.
---	---

Details

For the linear regression biplot handled by this method, let $\mathbf{X} \in \mathbb{R}^{n \times p}$ denote the *processed* data matrix stored in the biplot object after centring and any optional scaling performed by biplot(), and let $\mathbf{Z} \in \mathbb{R}^{n \times 2}$ denote the externally supplied display coordinates of the n samples. Write $\mathbf{Z} = [\mathbf{z}_1 \ \mathbf{z}_2]$, where $\mathbf{z}_1$ and $\mathbf{z}_2$ are the first and second displayed coordinates respectively. In contrast to a PCA biplot, the sample map is taken as given and the variable axes are then fitted to that map by multivariate least squares. This is the regression-biplot point of view used in the biplot literature for general low-dimensional sample maps (Gower and Hand, 1996; Gower, Lubbe and le Roux, 2011).

The fitted linear model is

$$\mathbf{X} = \mathbf{Z}\mathbf{H}^\top + \mathbf{E},$$

where, when $\mathbf{Z}$ has full column rank,

$$\mathbf{H}^\top = (\mathbf{Z}^\top \mathbf{Z})^{-1} \mathbf{Z}^\top \mathbf{X}.$$

Hence the fitted values are

$$\hat{\mathbf{X}} = \mathbf{Z}\mathbf{H}^\top = \mathbf{Z}(\mathbf{Z}^\top \mathbf{Z})^{-1} \mathbf{Z}^\top \mathbf{X} = \mathbf{P}_Z \mathbf{X},$$

where $\mathbf{P}_Z$ is the orthogonal projector onto the column space of $\mathbf{Z}$. More generally, if the supplied display coordinates are rank-deficient, the same fitted matrix $\hat{\mathbf{X}}$ is obtained by interpreting $\mathbf{P}_Z$ as the orthogonal projector onto $\text{col}(\mathbf{Z})$. The regression biplot therefore displays the variables through the least-squares predictions obtained from the supplied 2D sample map (Gower and Hand, 1996; Gower, Lubbe and le Roux, 2011).

If $\mathbf{h}_{(j)}$ denotes the j th column of $\mathbf{H}$, then the predicted value of variable j for sample i is

$$\hat{x}_{ij} = \mathbf{z}_i^\top \mathbf{h}_{(j)}.$$

The calibrated axis for variable j has direction $\mathbf{h}_{(j)}$, and the point on that axis corresponding to marker value μ is

$$\mathbf{p}_{\mu j} = \frac{\mu}{\mathbf{h}_{(j)}^\top \mathbf{h}_{(j)}} \mathbf{h}_{(j)}.$$

This is the calibration formula used to place tick marks and to recover predicted values from projections onto the displayed axis, in direct analogy with calibrated-axis biplot constructions (Gabriel, 1971; Gower, Lubbe and le Roux, 2011). All such predicted values are on the same centred/scaled scale as the stored matrix $\mathbf{X}$; if needed, they can be back-transformed to the original variable scale using the means and standard deviations stored in the input biplot object.

A regression biplot admits a natural family of *predictivity* measures on the variable side. Let $\mathbf{x}_{(j)}$ denote column j of $\mathbf{X}$, let $\hat{\mathbf{x}}_{(j)}$ denote column j of $\hat{\mathbf{X}}$, and let $\mathbf{e}_{(j)} = \mathbf{x}_{(j)} - \hat{\mathbf{x}}_{(j)}$. Since $\hat{\mathbf{X}} = \mathbf{P}_Z \mathbf{X}$ is an orthogonal projection, the residual matrix satisfies

$$\hat{\mathbf{X}}^\top \mathbf{E} = \mathbf{0},$$

and therefore

$$\mathbf{X}^\top \mathbf{X} = \hat{\mathbf{X}}^\top \hat{\mathbf{X}} + (\mathbf{X} - \hat{\mathbf{X}})^\top (\mathbf{X} - \hat{\mathbf{X}}).$$

This is the variable-side, or Type B, orthogonality that justifies variance-accounted-for ratios for the columns of $\mathbf{X}$; it is the same side of the orthogonality argument that underlies column-wise predictivities in the biplot literature (Gower, Lubbe and le Roux, 2011; Greenacre, 2010).

The predictivity of variable j is therefore defined by

$$\phi_j = \frac{\|\hat{\mathbf{x}}_{(j)}\|^2}{\|\mathbf{x}_{(j)}\|^2} = 1 - \frac{\|\mathbf{x}_{(j)} - \hat{\mathbf{x}}_{(j)}\|^2}{\|\mathbf{x}_{(j)}\|^2}, \quad j = 1, \dots, p.$$

Thus ϕ_j is the proportion of the sum of squares of variable j reproduced by the regression biplot, equivalently the ordinary multiple-regression R^2 obtained by regressing variable j on the displayed coordinates $\mathbf{Z}$. Each ϕ_j lies in $[0, 1]$; values near one indicate that the variable is well predicted by the displayed map, while values near zero indicate that the variable is poorly reproduced by the chosen display (Greenacre, 2010).

A natural overall quality-of-display measure is the proportion of total sum of squares reproduced by the display,

$$R_{\text{disp}}^2 = \frac{\|\hat{\mathbf{X}}\|_F^2}{\|\mathbf{X}\|_F^2} = 1 - \frac{\|\mathbf{X} - \hat{\mathbf{X}}\|_F^2}{\|\mathbf{X}\|_F^2}.$$

Because the column-wise decomposition above is orthogonal, this overall quality can be written as a weighted average of the variable predictivities:

$$R_{\text{disp}}^2 = \sum_{j=1}^p w_j \phi_j,$$

where

$$w_j = \frac{\|\mathbf{x}_{(j)}\|^2}{\|\mathbf{X}\|_F^2}, \quad \sum_{j=1}^p w_j = 1.$$

Hence variables with larger sums of squares contribute more to the overall quality. In particular, if the original call to `biplot()` used `scale = TRUE`, so that all processed variables have equal sums of squares, then the weights are equal and

$$R_{\text{disp}}^2 = \frac{1}{p} \sum_{j=1}^p \phi_j.$$

This weighted-average interpretation is often the most natural way to read the overall regression-biplot quality, since it combines the separate variable predictivities into a single display-wide summary (Greenacre, 2010).

The quantities ϕ_j and R_{disp}^2 depend only on the fitted projection $\mathbf{P}_Z \mathbf{X}$ and therefore only on the subspace $\text{col}(\mathbf{Z})$. They do *not* depend on any particular basis chosen for that subspace. In particular, the variable predictivities ϕ_j do not require any QR decomposition.

To decompose the total display quality into separate contributions for the two displayed dimensions, this package applies an *ordered orthogonalization* of the supplied display coordinates. Specifically, define

$$\mathbf{u}_1 = \mathbf{z}_1, \quad \mathbf{q}_1 = \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|}$$

whenever $\mathbf{u}_1 \neq \mathbf{0}$, and then define

$$\mathbf{u}_2 = \mathbf{z}_2 - \mathbf{q}_1 \mathbf{q}_1^\top \mathbf{z}_2, \quad \mathbf{q}_2 = \frac{\mathbf{u}_2}{\|\mathbf{u}_2\|}$$

whenever $\mathbf{u}_2 \neq \mathbf{0}$. Equivalently, $\mathbf{Q} = [\mathbf{q}_1 \ \mathbf{q}_2]$ is obtained from the QR decomposition of $\mathbf{Z}$, preserving the supplied column order. The vectors $\mathbf{q}_1$ and $\mathbf{q}_2$ are orthonormal and span the same display subspace as the nonzero columns of $\mathbf{Z}$.

Because $\mathbf{Q}$ and $\mathbf{Z}$ span the same subspace, the orthogonal projector may also be written as

$$\mathbf{P}_Z = \mathbf{Q} \mathbf{Q}^\top.$$

Consequently,

$$\hat{\mathbf{X}} = \mathbf{Q} \mathbf{Q}^\top \mathbf{X} = \mathbf{q}_1 \mathbf{q}_1^\top \mathbf{X} + \mathbf{q}_2 \mathbf{q}_2^\top \mathbf{X}$$

whenever both orthogonalized directions are present. Since $\mathbf{q}_1^\top \mathbf{q}_2 = 0$, the two fitted parts are orthogonal and their sums of squares add. This yields the dimension-specific contributions

$$R_1^2 = \frac{\|\mathbf{q}_1 \mathbf{q}_1^\top \mathbf{X}\|_F^2}{\|\mathbf{X}\|_F^2},$$

and

$$R_{2|1}^2 = \frac{\|\mathbf{q}_2 \mathbf{q}_2^\top \mathbf{X}\|_F^2}{\|\mathbf{X}\|_F^2},$$

so that

$$R_{\text{disp}}^2 = R_1^2 + R_{2|1}^2$$

whenever the display space is two-dimensional.

Care should be taken when interpreting this decomposition. If the columns of $\mathbf{Z}$ are already orthogonal, then the two displayed contributions correspond directly to the first and second supplied display axes. If the columns of $\mathbf{Z}$ are not orthogonal, however, the decomposition is *ordered*. The

first contribution R_1^2 is attributable to the first supplied display coordinate $\mathbf{z}_1$. The second contribution $R_{2|1}^2$ is attributable to the component of the second supplied display coordinate $\mathbf{z}_2$ that is orthogonal to the first. Thus $R_{2|1}^2$ should be interpreted as the additional contribution of “Dim 2 given Dim 1”, not as the contribution of the raw second column of $\mathbf{Z}$ considered in isolation. The ordering of the columns of $\mathbf{Z}$ is therefore important for this decomposition.

The same ordered orthogonalization yields a decomposition of each variable’s predictivity:

$$\phi_j = \phi_{j1} + \phi_{j,2|1},$$

where

$$\phi_{j1} = \frac{\|\mathbf{q}_1 \mathbf{q}_1^\top \mathbf{x}_{(j)}\|^2}{\|\mathbf{x}_{(j)}\|^2} = \frac{(\mathbf{q}_1^\top \mathbf{x}_{(j)})^2}{\|\mathbf{x}_{(j)}\|^2},$$

and

$$\phi_{j,2|1} = \frac{\|\mathbf{q}_2 \mathbf{q}_2^\top \mathbf{x}_{(j)}\|^2}{\|\mathbf{x}_{(j)}\|^2} = \frac{(\mathbf{q}_2^\top \mathbf{x}_{(j)})^2}{\|\mathbf{x}_{(j)}\|^2}.$$

Thus ϕ_{j1} is the part of variable j ’s predictivity explained by the first supplied display dimension, while $\phi_{j,2|1}$ is the additional part explained by the second display dimension after removing its overlap with the first.

If the supplied display coordinates are collinear, then $\mathbf{u}_2 = \mathbf{0}$ and the effective display space is one-dimensional. In that case $R_{2|1}^2 = 0$ and $\phi_{j,2|1} = 0$ for all variables.

In addition to the sum-of-squares fit measures above, this method may also report direct-reading error diagnostics in the sense of Alves (2012). The purpose of these diagnostics is different from that of the predictivities ϕ_j . The quantities ϕ_j , ϕ_{j1} , $\phi_{j,2|1}$ and R_{disp}^2 measure how much of the variation in $\mathbf{X}$ is reproduced by the fitted regression biplot. By contrast, the Alves diagnostics measure how accurately values can be read directly from a displayed calibrated axis in the current two-dimensional map. Alves (2012) proposed this idea for predictive PCA biplots; in the present two-dimensional regression-biplot setting the same principle applies in a particularly simple form because there is only one displayed map.

For each sample $i = 1, \dots, n$ and variable axis $j = 1, \dots, p$, the reading taken from the displayed axis of variable j is precisely the fitted value

$$\hat{x}_{ij} = \mathbf{z}_i^\top \mathbf{h}_{(j)}.$$

The corresponding point on the calibrated axis is

$$\mathbf{p}_{ij} = \frac{\hat{x}_{ij}}{\mathbf{h}_{(j)}^\top \mathbf{h}_{(j)}} \mathbf{h}_{(j)},$$

obtained by substituting $\mu = \hat{x}_{ij}$ into the calibration formula. Thus the direct reading from the graph and the fitted value from the regression model coincide.

Let s_j denote the standard deviation used to standardize variable j . When `scale = TRUE`, the processed matrix $\mathbf{X}$ already has unit-variance columns and hence $s_j = 1$. The pointwise direct-reading error for sample i on variable axis j is defined by

$$\delta_{ij} = \frac{|x_{ij} - \hat{x}_{ij}|}{s_j}.$$

If the processed matrix is already standardized, then $\delta_{ij} = |x_{ij} - \hat{x}_{ij}|$; in that case δ_{ij} is the direct analogue of Alves’s standard predictive error. More generally, dividing by s_j expresses the

discrepancy on a comparable variable-wise scale. The quantity δ_{ij} is therefore a sample-by-axis direct-reading error.

The corresponding axis-level mean direct-reading error is

$$\bar{\delta}_j = \frac{1}{n} \sum_{i=1}^n \delta_{ij} = \frac{1}{n} \sum_{i=1}^n \frac{|x_{ij} - \hat{x}_{ij}|}{s_j}.$$

This is the two-dimensional regression-biplot analogue of the mean standard predictive error of Alves (2012). Small values of $\bar{\delta}_j$ indicate that the calibrated axis for variable j supports accurate direct readings on average across the displayed observations, whereas large values indicate that direct readings from that axis are unreliable in the current display.

Let $\tau_{\text{axis}} > 0$ be a user-specified *tolerance parameter* for axis-level direct-reading error. Then the Alves selection rule specialized to the present two-dimensional regression biplot is

$$\text{retain axis } j \iff \bar{\delta}_j \leq \tau_{\text{axis}}.$$

Thus an axis is shown only when its average direct-reading error is at most the allowed tolerance. Larger values of τ_{axis} retain more axes and therefore produce denser displays; smaller values enforce stricter axis selection and lead to sparser, more conservative displays. In Alves (2012), values around 0.5 are discussed as a practical starting point in conventional settings, but no universal default should be assumed.

In addition to axis selection, Alves (2012) proposed a second *tolerance parameter* for individual sample-axis discrepancies. Let $\tau_{\text{units}} > 0$. A sample i is then flagged as an outlier with respect to axis j whenever

$$\delta_{ij} > \tau_{\text{units}}.$$

Such a flag indicates that, even if axis j is acceptable on average, the direct reading for sample i from that axis is poor in the current display. Alves (2012) discusses values around 0.75 as a practical starting point for this tolerance parameter, again subject to the application and the scale of the analysis.

Because this wrapper is tied to a single two-dimensional regression-biplot display, the quantities δ_{ij} and $\bar{\delta}_j$ are display-specific diagnostics. They are not measures of the quality of the underlying fitted subspace in the sum-of-squares sense; rather, they quantify the numerical accuracy of direct readings from the currently displayed axes. This distinction is central in Alves (2012), who emphasizes that direct-reading error is conceptually different from earlier axis-predictivity measures.

The Alves diagnostics and the regression-biplot predictivities are therefore complementary. The quantity ϕ_j is a variance-accounted-for ratio justified by Type B orthogonality and answers the question: “How much of variable j ’s sum of squares is reproduced by the displayed regression biplot?” The quantity $\bar{\delta}_j$ is a mean absolute direct-reading error and answers the different question: “How accurately can values of variable j be read from the displayed calibrated axis?” Consequently, a variable may have high ϕ_j and still have a non-negligible direct-reading error, while a variable with moderate ϕ_j may nevertheless support acceptable average direct readings. In this implementation, the ϕ_j -family is the primary set of sum-of-squares fit measures, whereas the Alves quantities $\bar{\delta}_j$ and δ_{ij} provide supplementary, display-specific diagnostics for axis selection and observation-level checking.

In contrast, a regression biplot does *not* in general satisfy the sample-side decomposition

$$\mathbf{X}\mathbf{X}^\top = \hat{\mathbf{X}}\hat{\mathbf{X}}^\top + (\mathbf{X} - \hat{\mathbf{X}})(\mathbf{X} - \hat{\mathbf{X}})^\top.$$

Consequently, PCA-style sample predictivities are not generally justified for a regression biplot. The principled sum-of-squares fit measures are the variable predictivities ϕ_j , the overall quality R^2_{disp} , and the ordered dimension-specific contributions described above, with the Alves direct-reading errors providing a distinct supplementary perspective on the quality of the displayed axes.

In the wrapped `bipl5_biplot` object, these formulas drive the bottom display-quality label, the hover-time predicted values $\hat{\mathbf{X}}$, and the calibrated linear axes stored in `mdsDisplay_12`. Since the regression display is tied to one externally supplied map, `wrap_bipl5.regress()` produces a single `mdsDisplay` only. There is no PC/CV toggle and no separate PCA-style sample-fit panel.

Value

An object of class `c("bipl5_biplot", "reg")`

References

- Gabriel, K. R. (1971). The biplot graphical display of matrices with application to principal component analysis. *Biometrika*, 58(3), 453–467. doi:[10.1093/biomet/58.3.453](https://doi.org/10.1093/biomet/58.3.453)
- Gower, J. C. and Hand, D. J. (1996). *Biplots*. London: Chapman & Hall.
- Gower, J. C., Lubbe, S. and le Roux, N. J. (2011). *Understanding Biplots*. Chichester: Wiley.
- Greenacre, M. (2010). *Biplots in Practice*. Bilbao: BBVA Foundation.
- la Grange, A., le Roux, N. and Gardner-Lubbe, S. (2009). BiplotGUI: Interactive Biplots in R. *Journal of Statistical Software*, 30(12), 1–37. doi:[10.18637/jss.v030.i12](https://doi.org/10.18637/jss.v030.i12)
- Alves, M. R. (2012). Evaluation of the predictive power of biplot axes to automate the construction and layout of biplots based on the accuracy of direct readings from common outputs of multivariate analyses: application to principal component analysis. *Journal of Chemometrics*, 26(5), 180–190. doi:[10.1002/cem.2433](https://doi.org/10.1002/cem.2433)

Examples

```
## Not run:
library(biplotEZ)
bp <- biplot(iris[, 1:4]) |>
  regress(Z = prcomp(iris[, 1:4])$x[, 1:2], group.aes = iris[, 5]) |>
  wrap_bipl5()
bp
plot(bp)

## End(Not run)
```

wrap_bipl5_gg

Render a biplotEZ biplot with ggplot2

Description

`wrap_bipl5_gg()` is the **ggplot2** counterpart of `wrap_bipl5`: it takes the same two-dimensional **biplotEZ** objects and returns an ordinary `ggplot` object instead of a reactive `plotly` widget. Sample points, class means, alpha bags and concentration ellipses are drawn as standard **ggplot2** layers, and the calibrated axes are drawn by `geom_calibrated_axis`.

Usage

```
wrap_bipl5_gg(
  x,
  axis_colour = "black",
  tick_colour = NULL,
  text_colour = NULL,
  label_colour = NULL,
  axis_linewidth = 0.25,
  axis_linetype = "solid",
  text_size = 2.6,
  label_size = 3.88,
  axis_labels = TRUE,
  axis_ticks = TRUE,
  axis_text = TRUE,
  tick_extend = TRUE,
  tick_length = 0.025,
  text_dodge = 0.03,
  label_dodge = 0.03,
  label_placement = c("positive", "negative", "peripheral"),
  point_size = 2,
  point_alpha = 0.6,
  sample_labels = NULL,
  class_means = NULL,
  alpha_bags = TRUE,
  conc_ellipses = TRUE,
  legend_title = "Group",
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  axis_percentages = TRUE,
  xlim = NULL,
  ylim = NULL,
  expand = TRUE,
  clip = "on",
  theme = ggplot2::theme_bw()
)
```

Arguments

<code>x</code>	A two-dimensional biplot object from biplotEZ , with <code>PCA()</code> , <code>CVA()</code> , <code>regress()</code> or <code>PCO()</code> applied.
<code>axis_colour</code>	Colour of the axis lines.
<code>tick_colour</code> , <code>text_colour</code> , <code>label_colour</code>	Colours of the tick marks, the marker values and the variable names; NULL follows <code>axis_colour</code> .
<code>axis_linewidth</code> , <code>axis_linetype</code>	Width and type of the axis lines.

text_size, label_size	Text sizes, in millimetres, of the marker values and the variable names.
axis_labels, axis_ticks, axis_text	Logical; whether to draw the variable names, the tick marks and the marker values.
tick_extend	Logical; regenerate markers along the whole visible axis from the recovered calibration. See Details.
tick_length	Length of the tick marks, as a proportion of the smaller of the panel width and height.
text_dodge, label_dodge	Orthogonal distances of the marker values and the variable names from the axis, as proportions of the smaller of the panel width and height.
label_placement	One of "positive" (the default; the increasing end of the axis), "negative", or "peripheral".
point_size, point_alpha	Size and opacity of the sample points.
sample_labels	Logical; whether to label the sample points. NULL (the default) follows the label setting of <code>biplotEZ::samples()</code> .
class_means	Logical; whether to draw class means. NULL (the default) follows the <code>class_means</code> flag of the biplot object.
alpha_bags, conc_ellipses	Logical; whether to draw alpha bags and concentration ellipses when the object carries them.
legend_title	Title of the group legend.
title, subtitle, caption	Plot annotations. <code>caption</code> defaults to the display-quality string that <code>wrap_bipl5</code> shows beneath the plotly biplot.
axis_percentages	Logical; append the percentage of variation explained to the two scaffolding-axis titles.
xlim, ylim, expand, clip	Passed to <code>coord_equal</code> .
theme	A ggplot2 theme added to the plot.

Details

The calibrated axes reproduce the geometry used by `gggda::geom_axis()`, the layer behind `ordr::geom_cols_axis()`: each axis is a line spanning the plotting window, tick marks are short segments orthogonal to that line, marker values are printed rotated to the angle of the axis and dodged to one side, and the variable name is written at the window border on the increasing end of the axis.

The two packages differ in where the markers come from. **gggda** derives them at draw time from the center and scale aesthetics using Wilkinson's extended breaks algorithm, which restricts it to variables whose calibration is a simple centring and scaling. **bipl5** instead consumes the calibrated coordinates that `biplotEZ::axes_coordinates()` has already computed, and recovers the affine

value-to-position map they imply. That map is then used to lay markers out against the final plotting window, so the marker values, their spacing and their phase are all exactly the ones **biplotEZ** chose, while the axes still span the panel and never inflate its range.

By default markers are regenerated along the whole visible axis (`tick_extend = TRUE`) using that recovered calibration, because the marker range **biplotEZ** returns is tied to the range of the fitted values and need not cover the corners of the plotting window. Set `tick_extend = FALSE` to draw only the markers `biplotEZ::axes_coordinates()` returned.

Axis element aesthetics accept a single value, one value per axis, or the string "biplotEZ" to adopt the aesthetics stored on the object by `biplotEZ::axes()`. `tick_colour`, `text_colour` and `label_colour` additionally accept `NULL`, in which case they follow `axis_colour`.

Principal coordinate analysis biplots built with spline axes are drawn as polylines, with each marker oriented by the local direction of the curve.

Value

A ggplot object, which can be further modified with the usual `+` syntax.

References

Gower, J. C. and Hand, D. J. (1996). *Biplots*. London: Chapman and Hall.

Gower, J. C., Lubbe, S. and le Roux, N. J. (2011). *Understanding Biplots*. Chichester: Wiley.

Brunson, J. C. and Read, Q. D. (2023). *ordr: A 'tidyverse' Extension for Ordinations and Biplots*. R package version 0.1.1.

See Also

[wrap_bipl5](#) for the interactive plotly renderer, and [geom_calibrated_axis](#) for the underlying layer.

Examples

```
## Not run:
library(biplotEZ)
library(ggplot2)

biplot(iris[, 1:4], scale = TRUE) |>
  PCA(group.aes = iris[, 5]) |>
  wrap_bipl5_gg(title = "Predictive biplot of Anderson iris measurements")

# the result is an ordinary ggplot, so it composes as usual
biplot(iris[, 1:4], scale = TRUE) |>
  PCA(group.aes = iris[, 5]) |>
  wrap_bipl5_gg(legend_title = "Species") +
  scale_color_brewer(type = "qual", palette = 2)

# CVA biplots carry class means
biplot(iris[, 1:4]) |>
  CVA(classes = iris[, 5]) |>
  wrap_bipl5_gg()
```

```
## End(Not run)
```

Index

aes, [10](#)
append_mdsDisplay, [2](#), [3](#)
append_mdsDisplay(), [7](#), [18](#)

biplot5, [3](#)
biplotEZ::CVA(), [18](#)
biplotEZ::PCA(), [18](#)
biplotEZ::PCO(), [18](#)
biplotEZ::regress(), [18](#)

colorpal, [4](#)
coord_equal, [36](#)

extract, [3](#), [4](#)

format_samples, [5](#)
format_samples(), [12](#)

geom_calibrated_axis, [9](#), [34](#), [37](#)
GeomCalibratedAxis
 (geom_calibrated_axis), [9](#)

init_biplot, [3](#), [12](#)
init_biplot(), [6](#), [8](#), [18](#), [19](#)

layer, [11](#)

overlay_fit, [13](#)

plot.bipl5_biplot, [3](#), [14](#), [21](#)
plot.bipl5_fit, [3](#), [14](#)
plot_ly, [20](#)
print.bipl5_biplot, [15](#)
print.bipl5_data, [16](#)
print.bipl5_fitmeasures, [16](#)
print.bipl5_mdsDisplay, [17](#)

remove_mdsDisplay, [3](#), [17](#)

scale_mds, [3](#), [18](#)
scale_mds(), [8](#), [12](#), [19](#)
score_axes, [19](#)

Symbol_List, [20](#)

wrap_bipl5, [3](#), [20](#), [34](#), [36](#), [37](#)
wrap_bipl5.CVA, [21](#)
wrap_bipl5.PCA, [21](#)
wrap_bipl5.PCO, [28](#)
wrap_bipl5.regress, [29](#)
wrap_bipl5_gg, [11](#), [12](#), [34](#)