

Package: moveEZ (via r-universe)

October 15, 2026

Title Animated Biplots

Version 1.3.1

Description Create animated biplots that enables dynamic visualisation of temporal or sequential changes in multivariate data by animating a single biplot across the levels of a time variable. It builds on objects from the 'biplotEZ' package, Lubbe S, le Roux N, Nienkemper-Swanepoel J, Ganey R, Buys R, Adams Z, Manefeldt P (2024) <doi:10.32614/CRAN.package.biplotEZ>, allowing users to create animated biplots that reveal how both samples and variables evolve over time.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

VignetteBuilder knitr

Depends R (>= 4.1.0)

Imports dplyr, biplotEZ, gganimate, ggplot2, GPABin, knitr, purrr, scales

Suggests testthat (>= 3.0.0), rmarkdown, tibble, RColorBrewer

Config/Needs/website rmarkdown

Config/testthat/edition 3

URL <https://muvisu.github.io/moveEZ/>

Config/roxygen2/version 8.1.0

Config/pak/sysreqs

libabsl-dev cmake libfreetype6-dev libgdal-dev gdal-bin libgeos-dev make libicu-dev libpng-dev libuv1-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev libx11-dev zlib1g-dev

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

Date/Publication 2026-10-15 10:07:26 UTC

RemoteUrl <https://github.com/muvisu/moveez>

RemoteRef HEAD

RemoteSha e8f8d28800213655c11f3ed6ce893e14e9122d14

Contents

.calibrate.axis	2
Africa_climate	3
Africa_climate_target	3
axes_moveEZ	4
check_vars_moveEZ	5
chull_moveEZ	5
evaluation	6
moveplot	8
moveplot2	10
moveplot3	12
Index	15

.calibrate.axis	<i>Calibrate axis</i>
-----------------	-----------------------

Description

Calibrate axis

Usage

```
.calibrate.axis(  
  j,  
  Xhat,  
  means,  
  sd,  
  axes.rows,  
  ax.which,  
  ax.tickvec,  
  ax.orthogxvec,  
  ax.orthogyvec  
)
```

Arguments

j	j
Xhat	Xhat
means	means
sd	sd
axes.rows	axes.rows
ax.which	ax.which
ax.tickvec	ax.tickvec
ax.orthogxvec	ax.orthogxvec
ax.orthogyvec	ax.orthogyvec

Value

Calibrated axes

Africa_climate	Climate studies example dataset
----------------	---------------------------------

Description

Data extracted from ERA5 hourly data on single levels from 1940 to present

Format

A dataset with 960 observations and 9 variables.

Details

Year 8 years from 1950 to 2020

Month 12 calendar months

Region 10 IPCC climate reference regions

AccPrec Accumulated precipitation

DailyEva Daily evaporation

Temp Mean temperature

SoilMois Soil moisture

SPI6 6-month standardised precipitation index

wind Windspeed

Source

DOI: 10.24381/cds.adbb2d47 (Accessed on 11-02-2025)

Africa_climate_target	Climate studies target example dataset
-----------------------	--

Description

Data extracted from ERA5 hourly data on single levels for 1989

Format

A dataset with 120 observations and 9 variables.

Details

- Year** 8 years from 1950 to 2020
- Month** 12 calendar months
- Region** 10 IPCC climate reference regions
- AccPrec** Accumulated precipitation
- DailyEva** Daily evaporation
- Temp** Mean temperature
- SoilMois** Soil moisture
- SPI6** 6-month standardised precipitation index
- wind** Windspeed

Source

DOI: 10.24381/cds.adbb2d47 (Accessed on 11-02-2025)

axes_moveEZ	<i>Provide axes coordinates</i>
-------------	---------------------------------

Description

Provide axes coordinates

Usage

```
axes_moveEZ(bp, which.var)
```

Arguments

- bp Object
- which.var which variable(s) to find coordinates

Value

Axes coordinates

check_vars_moveEZ	<i>Check the time and group variables</i>
-------------------	---

Description

Validates `time.var` and `group.var` before any plotting is attempted, so that unsuitable input stops with an informative error instead of failing downstream. Both must name factor columns of the data supplied to `biplotEZ::biplot()`, without missing values.

Usage

```
check_vars_moveEZ(bp, time.var, group.var)
```

Arguments

<code>bp</code>	biplot object from <code>biplotEZ</code>
<code>time.var</code>	time variable
<code>group.var</code>	group variable

Value

`bp`, with unused levels of `time.var` dropped from `bp$raw.X`.

chull_moveEZ	<i>Convex hulls per group</i>
--------------	-------------------------------

Description

Computes the convex hull of each group for a single level of the time variable. A hull needs at least three non-collinear observations; groups that do not meet this are left out of the hull data and returned separately so that they can be drawn as points instead.

Usage

```
chull_moveEZ(Y, group.var, group_levels)
```

Arguments

<code>Y</code>	tibble of sample coordinates (columns <code>V1</code> , <code>V2</code>) and grouping columns
<code>group.var</code>	group variable
<code>group_levels</code>	levels of <code>group.var</code> to construct hulls for

Value

A list with `hull`, the rows of `Y` that form the hull vertices of each group, and `points`, the rows of `Y` for groups with too few observations to construct a hull.

evaluation

Measures of comparison for move plot 3

Description

This function calculates measures of comparison after generalised orthogonal Procrustes Analysis is performed in `moveplot3`. Orthogonal Procrustes Analysis is used to compare a target to a testee configuration. The following measures are calculate: Procrustes Statistic (PS), Congruence Coefficient (CC), Absolute Mean Bias (AMB), Mean Bias (MB) and Root Mean Squared Bias (RMSB).

Usage

```
evaluation(bp, centring = TRUE)
```

Arguments

<code>bp</code>	biplot object from <code>moveEZ</code>
<code>centring</code>	logical argument to apply centring or not (default is TRUE)

Details

For `moveplot`, the function evaluates whether the fixed frame of `moveplot()`, where the variable vectors are computed once on all the data, is adequate, or whether a biplot should be constructed for each time slice with `moveplot2()` or `moveplot3()`. A PCA (or CVA) biplot is fitted to each level of the time variable and compared with the biplot of all the data, see Details.

For `moveplot`, the following are computed for each level of the time variable, in the order of its levels:

- the proportion of variance explained by the first dimension of the biplot of that time slice,
- the largest principal angle (in degrees) between the two dimensional subspace of the biplot of that time slice and that of the biplot of all the data. Zero means that the time slice spans the same plane,
- the loadings of the variables on the first dimension. The sign of each eigenvector is arbitrary, so the dimensions of each time slice are reflected to point in the same direction as those of the biplot of all the data (as `auto.align` in `moveplot2()`) before the loadings are compared.

A Spearman rank correlation test between the order of the time levels and each of these quantities tests for a monotone drift over time. A p-value of 0.05 or more fails to reject zero correlation: there is no evidence that the structure of the biplot changes over time, the fixed frame of `moveplot()` is adequate, and `moveplot2()` or `moveplot3()` are not needed. A p-value below 0.05 indicates a drift, and a biplot should be constructed for each time slice. The test only detects monotone trends; large but non-monotone changes are visible in the plots.

Value

<code>eval.tab</code>	Returns a table of the measures of comparison for each level of the time variable compared to the target.
<code>fit.plot</code>	Returns a line plot with the fit measures that are bounded between zero and one: PS and CC. A small PS value and large CC value indicate good fit.
<code>bias.plot</code>	Returns a line plot with bias measures that are unbounded: AMB, MB and RMSB. Small values indicate low bias.
<code>structure</code>	For <code>moveplot</code> , a data frame with the proportion of variance of the first dimension, the principal angle and the loadings on the first dimension for each level of the time variable.
<code>var.plot</code>	For <code>moveplot</code> , a line plot of the proportion of variance of the first dimension over time.
<code>angle.plot</code>	For <code>moveplot</code> , a line plot of the principal angle with the biplot of all the data over time.
<code>loadings.plot</code>	For <code>moveplot</code> , a line plot of the loadings of the variables on the first dimension over time.
<code>drift.tab</code>	For <code>moveplot</code> , a table of the Spearman rank correlations (ρ) of each quantity with time, their p-values and whether they are significant at the 5% level (*).

Examples

```

data(Africa_climate)
data(Africa_climate_target)
bp <- biplotEZ::biplot(Africa_climate, scaled = TRUE) |> biplotEZ::PCA()
results <- bp |> moveplot3(time.var = "Year", group.var = "Region", hulls = TRUE,
move = FALSE, target = NULL) |> evaluation()
results$eval.tab
results$fit.plot
results$bias.plot

data(Africa_climate)
data(Africa_climate_target)
bp <- biplotEZ::biplot(Africa_climate, scaled = TRUE) |> biplotEZ::PCA()
results <- bp |> moveplot3(time.var = "Year", group.var = "Region", hulls = TRUE,
move = FALSE, target = Africa_climate_target) |> evaluation()
results$eval.tab
results$fit.plot
results$bias.plot

# Should a biplot be constructed for each time slice?
bp <- biplotEZ::biplot(Africa_climate, scaled = TRUE) |> biplotEZ::PCA()
results <- bp |> moveplot(time.var = "Year", group.var = "Region", hulls = TRUE,
move = FALSE) |> evaluation()
results$drift.tab
results$var.plot
results$angle.plot
results$loadings.plot

```

moveplot

Move plot

Description

Create animated biplot on samples in a biplot

Usage

```
moveplot(
  bp,
  time.var,
  group.var,
  move = TRUE,
  hulls = TRUE,
  scale.var = 5,
  shadow = FALSE,
  which = NULL,
  label.vars = NULL
)
```

Arguments

bp	biplot object from biplotEZ
time.var	name of the time variable, given as a character string. Must be a factor column of the data supplied to biplot(); the order of its levels gives the order of the time slices and levels without observations are dropped. A numeric (e.g. integer year) or character column stops with an error, see Details.
group.var	name of the group variable, given as a character string. Must be a factor column of the data supplied to biplot().
move	whether to animate (TRUE) or facet (FALSE) samples, according to time.var
hulls	whether to display sample points or convex hulls. A hull requires at least three non-collinear observations per group per level of time.var; groups with fewer are displayed as points instead.
scale.var	scaling the vectors representing the variables
shadow	whether the animation will keep past states (only when hulls = FALSE)
which	integer index into the levels of group.var selecting which groups to display. Default (NULL) shows all groups.
label.vars	character vector of column name(s) used to label sample points, pasted together when more than one is supplied. Default (NULL) omits point labels.

Details

`time.var` and `group.var` must both be factors. `biplot()` treats every numeric column as a variable of the biplot, so a time variable stored as a number (e.g. an integer year) has to be converted with `factor()` before `biplot()` is called, not afterwards.

Missing values are handled by `biplot()`, which removes every row containing an NA in any column (including `time.var` and `group.var`) with a warning. The plot is constructed from the remaining rows.

Value

<code>bp</code>	Returns the elements of the biplot object <code>bp</code> from <code>biplotEZ</code> .
<code>plot</code>	An animated or a facet of biplots based on the dynamic frame.
<code>time.var, group.var</code>	The names of the time and group variables, used by <code>evaluation()</code> .

Examples

```
data(Africa_climate)
bp <- biplot(Africa_climate, scaled = TRUE) |> PCA()

# Convex hulls facet plot
bp |> moveplot(time.var = "Year", group.var = "Region", hulls = TRUE, move = FALSE)

# Samples facet plot
bp |> moveplot(time.var = "Year", group.var = "Region", hulls = FALSE, move = FALSE)

# Specifying colours with colour palette in biplotEZ

if(interactive()) {
  bp <- biplot(Africa_climate, scaled = TRUE, group.aes = Africa_climate$Region) |>
  PCA() |> samples(col = RColorBrewer::brewer.pal(10, "Paired"))
  bp |> moveplot(time.var = "Year", group.var = "Region", hulls = TRUE, move = FALSE)}

# Specifying plotting characters for grouping variable in biplotEZ

if(interactive()) {
  bp <- biplot(Africa_climate, scaled = TRUE, group.aes = Africa_climate$Region) |>
  PCA() |> samples(pch = c(19, 21, 3))
  bp |> moveplot(time.var = "Year", group.var = "Region", hulls = FALSE, move = FALSE)}

# Specifying opacity of plotting characters and size of variable labels

if(interactive()) {
  bp <- biplot(Africa_climate, scaled = TRUE, group.aes = Africa_climate$Region) |>
  PCA() |> samples(opacity = 0.4) |> axes(label.cex = 1.2)
  bp |> moveplot(time.var = "Year", group.var = "Region", hulls = FALSE, move = FALSE)}

# Specifying colours manually in biplotEZ

if(interactive()) {
```

```

bp <- biplot(Africa_climate, scaled = TRUE, group.aes = Africa_climate$Region) |>
PCA() |> samples(col = c("firebrick4", "indianred3", "tomato", "sandybrown",
  "khaki1", "palegreen1", "darkseagreen2", "mediumaquamarine", "deepskyblue4", "mediumpurple4"))
bp |> moveplot(time.var = "Year", group.var = "Region", hulls = TRUE, move = FALSE)}

# Extracting measures of fit of PCA
bp <- biplot(Africa_climate, scaled = TRUE) |> PCA() |> fit.measures()
bp <- bp |> moveplot(time.var = "Year", group.var = "Region", hulls = TRUE, move = FALSE)
bp$quality
bp$axis.predictivity

# Convex hulls move plot

if(interactive()) {
bp |> moveplot(time.var = "Year", group.var = "Region", hulls = TRUE, move = TRUE)}

# Samples move plot with shadows

if(interactive()) {
bp |> moveplot(time.var = "Year", group.var = "Region", hulls = FALSE, move = TRUE, shadow = TRUE)}

# CVA biplot
bp <- biplot(Africa_climate, scaled = TRUE) |> CVA(class = Africa_climate$Region)
bp <- bp |> moveplot2(group.var = "Region", time.var = "Year", move = FALSE)

# Extracting measures of fit of CVA
bp <- biplot(Africa_climate, scaled = TRUE) |>
  CVA(classes = Africa_climate$Region) |> fit.measures()
bp <- bp |> moveplot(time.var = "Year", group.var = "Region", hulls = TRUE, move = FALSE)
bp$quality
bp$axis.predictivity
bp$within.class.axis.predictivity
bp$within.class.sample.predictivity

```

moveplot2

Move plot 2

Description

Create animated biplot on samples and variables in a biplot

Usage

```

moveplot2(
  bp,
  time.var,
  group.var,
  move = TRUE,
  hulls = TRUE,
  scale.var = 5,

```

```

    auto.align = TRUE,
    align.time = NA,
    reflect = NA
  )

```

Arguments

<code>bp</code>	biplot object from <code>biplotEZ</code>
<code>time.var</code>	name of the time variable, given as a character string. Must be a factor column of the data supplied to <code>biplot()</code> ; the order of its levels gives the order of the time slices and levels without observations are dropped. A numeric (e.g. integer year) or character column stops with an error, see Details.
<code>group.var</code>	name of the group variable, given as a character string. Must be a factor column of the data supplied to <code>biplot()</code> .
<code>move</code>	whether to animate (TRUE) or facet (FALSE) samples and variables, according to <code>time.var</code>
<code>hulls</code>	whether to display sample points or convex hulls. A hull requires at least three non-collinear observations per group per level of <code>time.var</code> ; groups with fewer are displayed as points instead.
<code>scale.var</code>	scaling the vectors representing the variables
<code>auto.align</code>	whether to align the sign of the biplot axes of every time slice to the biplot of all the data (<code>bp</code>) automatically. Default (TRUE). See Details.
<code>align.time</code>	a vector specifying the levels of <code>time.var</code> for which the biplots should be reflected, to override <code>auto.align</code> where it does not give the desired orientation. The reflections are applied after <code>auto.align</code> ; set <code>auto.align = FALSE</code> to reflect the unaligned biplots instead. Default (NA) reflects none.
<code>reflect</code>	a character vector, of the same length as <code>align.time</code> , specifying the reflection to apply at each corresponding time point in <code>align.time</code> . One of "x" to reverse the direction of the x-axis (horizontal flip), "y" to reverse the direction of the y-axis (vertical flip) and "xy" for both. This follows <code>reflect.axis</code> of <code>biplotEZ::reflect()</code> . Default (NA) applies no reflection.

Details

`time.var` and `group.var` must both be factors. `biplot()` treats every numeric column as a variable of the biplot, so a time variable stored as a number (e.g. an integer year) has to be converted with `factor()` before `biplot()` is called, not afterwards.

The eigenvectors (PCA) or canonical vectors (CVA) of each time slice are only determined up to sign, so the biplot of one time slice can be a reflection of that of the next, which makes comparisons over time meaningless. With `auto.align = TRUE` each of the two dimensions of a time slice is reflected whenever its inner product with the corresponding dimension of the biplot of all the data in `bp` is negative. Reflections do not change the fit measures.

The sign of a dimension is decided by a small inner product when that dimension is nearly orthogonal to the corresponding dimension of `bp`, and the automatic alignment may then not give the desired orientation. `align.time` and `reflect` override it for the given levels of `time.var`. Time

slices whose dimensions swap order or rotate cannot be aligned by reflections, see [moveplot3\(\)](#) instead.

Missing values are handled by `biplot()`, which removes every row containing an NA in any column (including `time.var` and `group.var`) with a warning. The plot is constructed from the remaining rows.

Value

`bp` Returns the elements of the biplot object `bp` from `biplotEZ`.

`plot` An animated or a facet of biplots based on the dynamic frame.

Examples

```
data(Africa_climate)
bp <- biplot(Africa_climate, scaled = TRUE) |> PCA()

if(interactive()) {
  bp <- bp |> moveplot2(time.var = "Year", group.var = "Region", hulls = TRUE, move = TRUE)}

# Extracting measures of fit for PCA
bp <- bp |> moveplot2(time.var = "Year", group.var = "Region", hulls = TRUE, move = FALSE)

# Manual alignment instead of auto.align, reflect the x-axis at 1950

bp2 <- biplot(Africa_climate, scaled = TRUE) |> PCA() |>
  moveplot2(time.var = "Year", group.var = "Region", hulls = TRUE, move = FALSE,
    auto.align = FALSE, align.time = "1950", reflect = "x")
bp$quality
bp$axis.predictivity

# Extracting measures of fit for CVA
bp <- biplot(Africa_climate) |> CVA(classes = Africa_climate$Region)
bp <- bp |> moveplot2(time.var = "Year", group.var = "Region", hulls = TRUE, move = FALSE)
bp$quality
bp$axis.predictivity
bp$within.class.axis.predictivity
bp$within.class.sample.predictivity
```

moveplot3

Move plot 3

Description

Create animated biplot on samples and variables in a biplot with a given target

Usage

```
moveplot3(
  bp,
  time.var,
  group.var,
  move = TRUE,
  hulls = TRUE,
  scale.var = 5,
  target = NULL
)
```

Arguments

<code>bp</code>	biplot object from <code>biplotEZ</code>
<code>time.var</code>	name of the time variable, given as a character string. Must be a factor column of the data supplied to <code>biplot()</code> ; the order of its levels gives the order of the time slices and levels without observations are dropped. A numeric (e.g. integer year) or character column stops with an error, see Details.
<code>group.var</code>	name of the group variable, given as a character string. Must be a factor column of the data supplied to <code>biplot()</code> .
<code>move</code>	whether to animate (TRUE) or facet (FALSE) samples and variables, according to <code>time.var</code>
<code>hulls</code>	whether to display sample points or convex hulls. A hull requires at least three non-collinear observations per group per level of <code>time.var</code> ; groups with fewer are displayed as points instead.
<code>scale.var</code>	scaling the vectors representing the variables
<code>target</code>	Target data set to which all biplots should be matched consisting of the the same dimensions. If not specified, the centroid of all available biplot sample coordinates from <code>time.var</code> will be used. Default NULL.

Details

`time.var` and `group.var` must both be factors. `biplot()` treats every numeric column as a variable of the biplot, so a time variable stored as a number (e.g. an integer year) has to be converted with `factor()` before `biplot()` is called, not afterwards.

Missing values are handled by `biplot()`, which removes every row containing an NA in any column (including `time.var` and `group.var`) with a warning. The plot is constructed from the remaining rows. Since `moveplot3()` requires the same number of observations at every level of `time.var`, removed rows will usually cause it to stop; impute the missing values, or remove the affected samples at every time level, beforehand. The same applies to `target`, which may not contain missing values.

Value

<code>bp</code>	Returns the elements of the biplot object <code>bp</code> from <code>biplotEZ</code> .
<code>iter_levels</code>	The levels of the time variable.

coord_set	The coordinates of the configurations before applying Generalised Orthogonal Procrustes Analysis.
GPA_list	The coordinates of the configurations after applying Generalised Orthogonal Procrustes Analysis.
plot	An animated or a facet of biplots based on the dynamic frame.

Examples

```
data(Africa_climate)
data(Africa_climate_target)
bp <- biplot(Africa_climate, scaled = TRUE) |> PCA()
bp |> moveplot3(time.var = "Year", group.var = "Region", hulls = TRUE,
move = FALSE, target = NULL)

if(interactive()) {
bp |> moveplot3(time.var = "Year", group.var = "Region", hulls = TRUE,
move = TRUE, target = NULL)}
bp |> moveplot3(time.var = "Year", group.var = "Region", hulls = TRUE,
move = FALSE, target = Africa_climate_target)
```

Index

* **datasets**

- Africa_climate, [3](#)
- Africa_climate_target, [3](#)
- .calibrate.axis, [2](#)

- Africa_climate, [3](#)
- Africa_climate_target, [3](#)
- axes_moveEZ, [4](#)

- check_vars_moveEZ, [5](#)
- chull_moveEZ, [5](#)

- evaluation, [6](#), [9](#)

- moveplot, [8](#)
- moveplot2, [10](#)
- moveplot3, [12](#), [12](#)