Package 'animbook'

October 30, 2023

Title Visualizing changes in performance measures and demographic affiliations using animation

Version 0.0.0.9000

Description Create an interactive visualization to be used for communication purposes. Providing the function for preparing, plotting, and animating the data.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

URL https://github.com/KrisanatA/animbook

BugReports https://github.com/KrisanatA/animbook/issues

Depends R (>= 2.10)

LazyData true

Suggests knitr,
rmarkdown

VignetteBuilder knitr

Imports dplyr,
gganimate,
ggplot2,
plotly,
purrr,
RColorBrewer,
rlang,
stats,
tibble,
tidyr,
tidyselect

R topics documented:

`aeles` *Australian election study data*

Description

The aes dataset contains the answer of the surveys which is done in 2019 for the election. This dataset only includes the id, year, party, gender from the the full survey data. The year column come from the transformations of two different question to see whether the voter voted the same party in 2016 and 2019 or not and if not who did they voted for before.

Usage

```
aeles
```

Format

A data frame with 1,468 rows and 4 variables

id The id of the respondent

year Year

party Party that the respondent vote for in the House of Representatives

gender Gender of the respondent

Source

This dataset is from the following; Australian Election Study https://dataverse.ada.edu.au/file.xhtml?fileId=18013&version=3.0

`anim_animate` *Modified the ggplot object*

Description

This function by default will modified the ggplot object before the user can pass it to the rendering of choice.

Usage

```
anim_animate(plot, modify = FALSE)
```

Arguments

`plot`	ggplot object
`modify`	Default is FALSE which supplied the function needed for the rendering package.

Value

A gganimate object if the rendering is gganimate or a ggplot object if the rendering is plotly.

Examples

```
animbook <- anim_prep(data = osiris, id = ID, values = sales, time = year, color = japan)

plot <- wallaby_plot(animbook)

animate <- anim_animate(plot)

plotly::ggplotly(animate)
```

---

anim_prep *Prepare Numerical Data for Visualizations*

Description

This function prepares the numerical data into the format the plot function required by assigning frames, grouping data, scaling values, and creating necessary data and settings for the plot function.

Usage

```
anim_prep(
  data,
  id = NULL,
  values = NULL,
  time = NULL,
  label = NULL,
  ngroup = 5L,
  breaks = NULL,
  group_scaling = NULL,
  color = NULL,
  time_dependent = TRUE,
  scaling = "rank",
  runif_min = 1,
  runif_max = 50
)
```

Arguments

data	A data frame containing the data to be prepared for visualization.
id	The column name that represents the unique identifier variable.
values	The column name that contains the numeric values to be visualized.
time	The column name represents the time variable.
label	A vector of labels to be used for the y-axis in the visualization.
ngroup	The number of groups or categories to create for scaling values.
breaks	A vector of breaks for creating bins.

group_scaling	The column name that represents the grouping variable.
color	The column name to used in ggplot2::aes() for the plot function.
time_dependent	Logical. Should the visualization be time-dependent? Default is TRUE.
scaling	The scaling method to be used; "rank" or "absolute."
runif_min	The minimum value for random addition to frame numbers.
runif_max	The maximum value for random addition to frame numbers.

Details

The function takes the input data and performs several operations to prepare it for visualizations. It assigns frames, groups data, scales values, and creates necessary data and settings for the plot function.

Value

An animbook object:

data	A data frame with prepared data for visualization.
settings	A list of settings to be used in plot function, including gap, xbreaks, label, scaling, time_dependent, runif_min, and runif_max.

Examples

```
anim_prep(data = osiris, id = ID, values = sales, time = year)

anim_prep(data = osiris, id = ID, values = sales, time = year,
group_scaling = country)

anim_prep(data = osiris, id = ID, values = sales, time = year,
scaling = "absolute")

anim_prep(data = osiris, id = ID, values = sales, time = year,
group_scaling = country, scaling = "absolute")
```

---

anim_prep_cat *Prepare Category Data for Visualizations*

Description

This function prepares the category data into the format the plot function required by assigning frames and creating necessary data and settings for the plot function.

Usage

```
anim_prep_cat(
  data,
  id = NULL,
  values = NULL,
  time = NULL,
  label = NULL,
```

```
  order = NULL,
  color = NULL,
  time_dependent = TRUE,
  runif_min = 1,
  runif_max = 50
)
```

Arguments

data	A data frame containing the category values to be prepared for visualization.
id	The column name that represents the unique identifier variable.
values	The column name that contains the categorical values to be visualized.
time	The column name represents the time variable.
label	A vector of labels to be used for the y-axis in the visualization.
order	A vector of order for sorting the category values.
color	The column name to be used in `ggplot2::aes()` for the plot function.
time_dependent	Logical. Should the visualization be time-dependent? Default is TRUE.
runif_min	The minimum value for random addition to frame numbers.
runif_max	The maximum value for random addition to frame numbers.

Details

The function takes the input data and performs several operations to prepare it for visualizations. It assigns frames and creates necessary data and settings for the plot function.

Value

An animbook object:

data	A data frame with prepared data for visualization.
settings	A list of settings to be used in plot function, including gap, xbreaks, label, scaling, time_dependent, runif_min, and runif_max

Examples

```
anim_prep_cat(data = aeles, id = id, values = party, time = year)
```

---

cat_change *Simulated data with some change (category)*

Description

This data has changes from category A to E between two time points.

Usage

```
cat_change
```

Format

A data frame with 400 rows and 4 variables

id The id of the organisation

time time

gp Either X or Y

qnt Quantile group for the two times

Examples

```
d <- anim_prep_cat(cat_change, id = id, values = qnt,
time = time, color = gp, time_dependent = FALSE)

d_p <- wallaby_plot(d, height = 1)

d_p_anim <- anim_animate(d_p)
```

---

dbl_change *Simulated data with some change (numerical)*

Description

This data contained the numerical values for each observation.

Usage

```
dbl_change
```

Format

A data frame with 400 rows and 4 variables

id The id of the organisation

time time

gp Either X or Y

values Numerical values represent sales

Examples

```
d <- anim_prep(dbl_change, id = id, values = values,
time = time, color = gp, time_dependent = FALSE)

d_p <- wallaby_plot(d, height = 1)

d_p_anim <- anim_animate(d_p)
```

funnel_web_plot *Turn the data into a facetted plot*

Description

This function takes in the data which has been prepared by either anim_prep() or anim_prep_cat() and return the ggplot object. The user can still modify the plot the same as normal using the ggplot2 function.

Usage

```
funnel_web_plot(
  object,
  palette = RColorBrewer::brewer.pal(9, "Set1"),
  rendering = "ggplot",
  ...
)
```

Arguments

object	The animbook object returned from the prep function.
palette	The vector of the palette used by the function to supply the color to each group.
rendering	The choice of method used to create and display the plot, either gganimate or plotly.
...	Additional arguments for customization, see details for more information.

Details

This function takes prepared data and generates a ggplot object. The funnel web plot is the plot that shows the line facetted plot showing the pattern between time period. The line jitter can be controlled using additional arguments such as height and width to control the appearance. For the shading area, the alpha argument can be used.

Value

Return a ggplot object

Examples

```
animbook <- anim_prep(data = osiris, id = ID, values = sales, time = year, color = japan)

funnel_web_plot(animbook)
```

kangaroo_plot *Turn the data into a ggplot object for the animate function*

Description

This function takes in the data which has been prepared by the anim_prep() or anim_prep_cat() and return the ggplot object. The user can still modify the plot the same as normal.

Usage

```
kangaroo_plot(
  object,
  palette = RColorBrewer::brewer.pal(9, "Set1"),
  rendering = "ggplot",
  ...
)
```

Arguments

object	The animbook object returned from the prep function.
palette	The vector of the palette used by the function to supply the color to each group.
rendering	The choice of method used to create and display the plot, either gganimate or plotly.
...	Additional arguments for customization, see details for more information.

Details

This function takes prepared data and generates a ggplot object. The kangaroo plot is the plot that shows the movement between groups over time. The jitter point can be controlled using additional arguments such as height, width, and size to control the appearance. For the shading area, the alpha argument can be used.

Value

Return a ggplot object

Examples

```
animbook <- anim_prep(data = osiris, id = ID, values = sales, time = year, color = japan)

kangaroo_plot(animbook)
```

osiris *Osiris firm sales data*

Description

The Osiris dataset contains information on listed, and major unlisted/delisted companies across the world from 2006 to 2018. This dataset only includes the year, ID, country, sales, and japan variables from the full Osiris dataset to give the user an example of the dataset format.

Usage

```
osiris
```

Format

A data frame with 10,270 rows and 5 variables

year Year

ID BvD(Bureau van Dijk) ID

country Address of incorp. - Country

sales Sales

japan Whether the firm is froom Japan or not

Source

This dataset is from the following; Bureau van Dijk https://www.bvdinfo.com/en-gb/our-products/data/international/osiris.

wallaby_plot *Turn the data into a Sankey flow plot for animate function*

Description

This function takes in the data which has been prepared by the anim_prep() or anim_prep_cat() and return the ggplot object. The user can still modify the plot the same as normal ggplot.

Usage

```
wallaby_plot(
  object,
  group_palette = RColorBrewer::brewer.pal(9, "Set1"),
  shade_palette = RColorBrewer::brewer.pal(9, "Set1"),
  rendering = "ggplot",
  subset = "top",
  relation = "one_many",
  total_point = NULL,
  ...
)
```

Arguments

object	The animbook object returned from the prep function
group_palette	The vector of the palette used by the function to supply the color to each group.
shade_palette	The vector of the palette used by the function to supply the color to the shaded area.
rendering	The choice of method used to create and display the plot, either gganimate or plotly.
subset	A character string specifying the variable used for subsetting the data. The "top" and "bottom" strings can also be used in this argument.
relation	The choice of relationship for the values to display on the plot, either "one_many." or "many_one."
total_point	The number of points the users want for the wallaby plot. Default is NULL, the number of the point is equal to the original number of points.
...	Additional arguments for customization, see details for more information.

Details

This function takes prepared data and generates a ggplot object. The wallaby plot is the Sankey flow plot that shows the movement of the subset data. The point position and point size in the shaded area can be controlled using additional arguments such height, width, and size. For the shading area, the alpha argument can be used.

Value

Return a ggplot object

Examples

```
animbook <- anim_prep(data = osiris, id = ID, values = sales, time = year, color = japan)

wallaby_plot(animbook)
```

Index