

Package: `pacta.multi.loanbook.plot` (via `r-universe`)

September 18, 2024

Title Tools to Visualize Climate Metrics for Multiple Loanbooks

Version 0.0.0.9005

Description This is an experimental package that contains functions to help plot aggregated PACTA results, with a focus on making it easy to navigate and understand results across a large number of banks and loan books.

License MIT + file LICENSE

URL <https://github.com/RMI-PACTA/pacta.multi.loanbook.plot>

BugReports <https://github.com/RMI-PACTA/pacta.multi.loanbook.plot/issues>

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Config/Needs/website rmi-pacta/pacta.pkgdown.rmitemplate

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Imports dplyr, ggplot2, glue, magrittr, networkD3, plotly, r2dii.plot, rlang, scales, tidyr, webshot

Repository <https://rmi-pacta.r-universe.dev>

RemoteUrl <https://github.com/rmi-pacta/pacta.multi.loanbook.plot>

RemoteRef HEAD

RemoteSha dd14a145ac61b3b0b6446612742b51a9926c9125

Contents

<code>plot_sankey</code>	2
<code>plot_scatter</code>	3
<code>plot_scatter_alignment_exposure</code>	4
<code>plot_scatter_animated</code>	5

plot_timeline	6
prep_sankey	7
prep_scatter	8
prep_scatter_alignment_exposure	9
prep_scatter_animated	10
prep_timeline	11

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

plot_sankey	<i>Make a sankey plot</i>
-------------	---------------------------

Description

Make a sankey plot

Usage

```
plot_sankey(
  data,
  group_var,
  capitalise_node_labels = TRUE,
  save_png_to = NULL,
  png_name = "sankey.png",
  nodes_order_from_data = FALSE
)
```

Arguments

data	data.frame. Should have the same format as output of prep_sankey() and contain columns: "middle_node", optionally "middle_node2", "is_aligned", "loan_size_outstanding", and any column implied by group_var.
group_var	Character. Vector of length 1. Variable to group by.
capitalise_node_labels	Logical. Flag indicating if node labels should be converted into better looking capitalised form.
save_png_to	Character. Path where the output in png format should be saved
png_name	Character. File name of the output.
nodes_order_from_data	Logical. Flag indicating if nodes order should be determined by an algorithm (in case of big datasets often results in a better looking plot) or should they be ordered based on data.

Examples

```
# TODO
```

plot_scatter	<i>Plot alignment scatterplot</i>
--------------	-----------------------------------

Description

Plot alignment scatterplot

Usage

```
plot_scatter(
  data,
  sector = NULL,
  scenario_source = NULL,
  scenario = NULL,
  year = NULL,
  region = NULL,
  title = NULL,
  subtitle = NULL,
  alignment_limit = NULL,
  data_level = c("company", "group_var"),
  cap_outliers = NULL,
  floor_outliers = NULL
)
```

Arguments

data	data.frame. Should have the same format as output of prep_scatter() and contain columns: 'name', 'buildout', 'phaseout', 'net'.
sector	Character. Sector name to be used in the plot title.
scenario_source	Character. Scenario source to be used in the plot caption.
scenario	Character. Scenario name to be used in the plot caption.
year	Integer. Year of the analysis to be used in the plot caption.
region	Character. Region to be used in the plot caption.
title	Character. Custom title if different than default.
subtitle	Character. Custom subtitle if different than default.
alignment_limit	Numeric. Limit to be applied to the x- and y-axis scales and to alignment values for colouring. By default the maximum absolute alignment value of is used.
data_level	Character. Level of the plotted data. Can be 'group_var' or 'company'.
cap_outliers	Numeric. Cap which should be applied to the alignment values in the data. Values bigger than cap are plotted on the border of the plot.
floor_outliers	Numeric. Floor which should be applied to the alignment values in the data. Values smaller than floor are plotted on the border of the plot.

Value

object of type "ggplot"

Examples

```
# TODO
```

```
plot_scatter_alignment_exposure  
  Plot alignment scatterplot
```

Description

Plot alignment scatterplot

Usage

```
plot_scatter_alignment_exposure(  
  data,  
  floor_outliers,  
  cap_outliers,  
  group_var,  
  currency  
)
```

Arguments

data	data.frame. Should have the same format as output of prep_scatter() and contain columns: 'name', 'buildout', 'phaseout', 'net', and any column implied by group_var.
floor_outliers	Numeric. Floor which should be applied to the alignment values in the data. Values smaller than floor are plotted on the border of the plot.
cap_outliers	Numeric. Cap which should be applied to the alignment values in the data. Values bigger than cap are plotted on the border of the plot.
group_var	Character. Character specifying the variable that contains the groups by which to analyse the loan books.
currency	Character. Currency to display in the plot labels.

Value

object of type "ggplot"

Examples

```
# TODO
```

 plot_scatter_animated *Plot alignment scatterplot*

Description

Plot alignment scatterplot

Usage

```
plot_scatter_animated(
  data,
  data_level = c("company", "group_var"),
  sector = NULL,
  scenario_source = NULL,
  scenario = NULL,
  region = NULL,
  title = NULL,
  subtitle = NULL,
  alignment_limit = NULL,
  cap_outliers = NULL,
  floor_outliers = NULL
)
```

Arguments

data	data.frame. Should have the same format as output of prep_scatter_animated() and contain columns: 'name', 'buildout', 'phaseout', 'net' and 'year'.
data_level	Character. Level of the plotted data. Can be 'group_var' or 'company'.
sector	Character. Sector name to be used in the plot title.
scenario_source	Character. Scenario source to be used in the plot caption.
scenario	Character. Scenario name to be used in the plot caption.
region	Character. Region to be used in the plot caption.
title	Character. Custom title if different than default.
subtitle	Character. Custom subtitle if different than default.
alignment_limit	Numeric. Limit to be applied to the x- and y-axis scales and to alignment values for colouring. By default the maximum absolute alignment value from data is used.
cap_outliers	Numeric. Cap which should be applied to the alignment values in the data. Values bigger than cap are plotted on the border of the plot.
floor_outliers	Numeric. Floor which should be applied to the alignment values in the data. Values smaller than floor are plotted on the border of the plot.

Value

object of type "plotly"

Examples

```
# TODO
```

plot_timeline	<i>Plot alignment timeline</i>
---------------	--------------------------------

Description

Plot alignment timeline

Usage

```
plot_timeline(  
  data,  
  sector = NULL,  
  scenario_source = NULL,  
  scenario = NULL,  
  region = NULL,  
  group_var = NULL,  
  title = NULL,  
  subtitle = NULL,  
  alignment_limits = NULL  
)
```

Arguments

data	data.frame Should have the same format as output of prep_timeline() and contain columns: 'direction', 'year', 'exposure_weighted_net_alignment', and any column implied by group_var.
sector	Character. Sector name to be used in the plot title.
scenario_source	Character. Scenario source to be used in the plot caption.
scenario	Character. Scenario name to be used in the plot caption.
region	Character. Region to be used in the plot caption.
group_var	Character. Vector of length 1. Variable to group by.
title	Character. Custom title if different than default.
subtitle	Character. Custom subtitle if different than default.
alignment_limits	Numeric vector of size 2. Limits to be applied to alignment values for colouring. By default maximum absolute value of 'exposure_weighted_net_alignment' is used.

Value

object of type "ggplot"

Examples

```
# TODO
```

prep_sankey	<i>Prepare data to plot using plot_sankey()</i>
-------------	---

Description

Prepare data to plot using plot_sankey()

Usage

```
prep_sankey(  
  data_alignment,  
  region,  
  year,  
  group_var,  
  middle_node,  
  middle_node2 = NULL  
)
```

Arguments

data_alignment	data.frame. Holds aggregated alignment metrics per company for tms sectors. Must contain columns: "name_abcd", "sector" and any column implied by group_var.
region	Character. Region to filter data_alignment data frame on.
year	Integer. Year on which data_alignment should be filtered.
group_var	Character. Vector of length 1. Variable to group by.
middle_node	Character. Column specifying the middle nodes to be plotted in sankey plot. Must be present in data_alignment.
middle_node2	Character. Column specifying the middle nodes to be plotted in sankey plot. Must be present in data_alignment.

Value

data.frame

Examples

```
# TODO
```

```
prep_scatter          Prepare data to plot scatterplot
```

Description

Prepare data to plot scatterplot

Usage

```
prep_scatter(
  data_bopo,
  data_net,
  data_level = c("group_var", "company"),
  year,
  sector,
  region,
  group_var,
  groups_to_plot = NULL
)
```

Arguments

data_bopo	data.frame. Data containing buildout and phaseout alignment values. Must contain columns: 'year', 'sector', 'region', 'direction' and either 'name_abcd' and 'alignment_metric' or 'exposure_weighted_net_alignment' plus any column implied by group_var.
data_net	data.frame. Data containing net alignment values. Must contain columns: group_var, 'year', 'sector', 'region', 'direction' and either 'name_abcd' and 'alignment_metric' or 'exposure_weighted_net_alignment'.
data_level	Character. Level of the plotted data. Can be 'group_var' or 'company'.
year	Integer. Year on which the data should be filtered.
sector	Character. Sector to filter data on.
region	Character. Region to filter data on.
group_var	Character. Vector of length 1. Variable to group by.
groups_to_plot	Character vector. Groups to filter on.

Value

data.frame

Examples

```
# TODO
```

```
prep_scatter_alignment_exposure
```

```
  Prepare data to plot using plot_scatter_alignment_exposure()
```

Description

Prepare data to plot using `plot_scatter_alignment_exposure()`

Usage

```
prep_scatter_alignment_exposure(  
  data,  
  year,  
  region,  
  scenario,  
  group_var,  
  exclude_groups = "benchmark"  
)
```

Arguments

<code>data</code>	<code>data.frame</code> . Holds net aggregated alignment metrics on the loan book level. Must contain columns: "scenario", "region", "sector", "year", "exposure_weighted_net_alignment", "sum_loan_size_outstanding" and any column implied by <code>group_var</code> .
<code>year</code>	Integer. Year on which data should be filtered.
<code>region</code>	Character. Region to filter data data frame on.
<code>scenario</code>	Character. Scenario to filter data data frame on.
<code>group_var</code>	Character. Vector of length 1. A column to group by. Must be available variables in data.
<code>exclude_groups</code>	Character. Character specifying any values from <code>group_var</code> that should not be included in the analysis. This is useful to remove benchmarks that are not meant to be compared at the same level. Defaults to "benchmark".

Value

`data.frame`

Examples

```
# TODO
```

prep_scatter_animated *Prepare data to plot animated scatterplot*

Description

Prepare data to plot animated scatterplot

Usage

```
prep_scatter_animated(  
  data_bopo,  
  data_net,  
  data_level = c("group_var", "company"),  
  sector,  
  region,  
  group_var,  
  groups_to_plot = NULL  
)
```

Arguments

data_bopo	data.frame. Data containing buildout and phaseout alignment values. Must contain columns: 'year', 'sector', 'region', 'direction' and either 'name_abcd' and 'alignment_metric' or 'exposure_weighted_net_alignment' plus any column implied by group_var.
data_net	data.frame. Data containing net alignment values. Must contain columns: group_var, 'year', 'sector', 'region', 'direction' and either 'name_abcd' and 'alignment_metric' or 'exposure_weighted_net_alignment'.
data_level	Character. Level of the plotted data. Can be 'group_var' or 'company'.
sector	Character. Sector to filter data on.
region	Character. Region to filter data on.
group_var	Character. Vector of length 1. Variable to group by.
groups_to_plot	Character vector. Groups to filter on.

Value

data.frame

Examples

```
# TODO
```

prep_timeline	<i>Prepare data to plot timeline</i>
---------------	--------------------------------------

Description

Prepare data to plot timeline

Usage

```
prep_timeline(data, sector, region, group_var, groups_to_plot)
```

Arguments

data	data.frame. Must contain columns: 'direction', 'year', 'exposure_weighted_net_alignment', 'sector' and any column implied by group_var.
sector	Character. Sector to filter data on.
region	Character. Region to filter data on.
group_var	Character. Vector of length 1. Variable to group by.
groups_to_plot	Character vector. Groups to filter on.

Value

data.frame

Examples

```
# TODO
```

Index

[plot_sankey](#), [2](#)
[plot_scatter](#), [3](#)
[plot_scatter_alignment_exposure](#), [4](#)
[plot_scatter_animated](#), [5](#)
[plot_timeline](#), [6](#)
[prep_sankey](#), [7](#)
[prep_scatter](#), [8](#)
[prep_scatter_alignment_exposure](#), [9](#)
[prep_scatter_animated](#), [10](#)
[prep_timeline](#), [11](#)