

Package: `pacta.aggregate.loanbook.plots` (via `r-universe`)

August 23, 2024

Title Functions to Help Interpret PACTA for Banks Results for Many Institutions

Version 0.0.0.9001

Description This is an experimental package that contains functions to help interpret PACTA for Banks 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

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports dplyr, ggplot2, glue, networkD3, r2dii.data, r2dii.plot, readr, rlang, tidyr, tibble, webshot, plotly, scales

Depends R (>= 2.10)

LazyData true

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

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

VignetteBuilder knitr

URL <https://rmi-pacta.github.io/pacta.aggregate.loanbook.plots/>

Config/testthat/edition 3

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

RemoteUrl <https://github.com/rmi-pacta/pacta.aggregate.loanbook.plots>

RemoteRef HEAD

RemoteSha 1bfa0892d36ca293cac5f62653654aa7a0b62326

Contents

abcd_test_data	2
activity_units	3
aggregate_alignment_loanbook_exposure	3
calculate_company_aggregate_alignment_sda	4
calculate_company_aggregate_alignment_tms	4
calculate_company_tech_deviation	5
create_benchmark_loanbook	6
loanbook_test_data	7
plot_sankey	7
plot_scatter	8
plot_scatter_animated	9
plot_timeline	10
prep_sankey	11
prep_scatter	12
prep_scatter_animated	13
prep_timeline	14
validate_data_has_expected_cols	15

Index	16
--------------	-----------

abcd_test_data	<i>ABCD test data</i>
----------------	-----------------------

Description

ABCD test data

Usage

abcd_test_data

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 582 rows and 13 columns.

activity_units	<i>Units for economic activities per sector and technology</i>
----------------	--

Description

Units for economic activities per sector and technology

Usage

```
activity_units
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 58 rows and 3 columns.

aggregate_alignment_loanbook_exposure	<i>Return loan book level aggregation of company alignment metrics by exposure</i>
---------------------------------------	--

Description

Return loan book level aggregation of company alignment metrics by exposure

Usage

```
aggregate_alignment_loanbook_exposure(data, matched, level = c("net", "bo_po"))
```

Arguments

data	data.frame. Holds output pf company indicators
matched	data.frame. Holds matched and prioritised loan book
level	Character. Vector that indicates if the aggregate alignment metric should be returned based on the net technology deviations (<code>net</code>) or disaggregated into build-out and phaseout technologies (<code>bo_po</code>).

calculate_company_aggregate_alignment_sda

Return company level sector alignment metric for each company

Description

Return company level sector alignment metric for each company

Usage

```
calculate_company_aggregate_alignment_sda(
  data,
  scenario_source = "geco_2021",
  scenario = "1.5c",
  time_frame = 5L
)
```

Arguments

data	data.frame. Holds the PACTA for Banks SDA results on company level.
scenario_source	Character. Vector that indicates which scenario_source to use for reference in the calculation of the alignment metrics. Currently, the only supported value is "geco_2021".
scenario	Character. Vector that indicates which scenario to calculate the alignment metric for. Must be a scenario available from scenario_source.
time_frame	Integer of length one. The number of forward looking years that should be considered in the analysis. Standard time_frame in PACTA is five years.

calculate_company_aggregate_alignment_tms

Return company level sector alignment metric for each company with option to disaggregate by buildout / phaseout.

Description

Return company level sector alignment metric for each company with option to disaggregate by buildout / phaseout.

Usage

```
calculate_company_aggregate_alignment_tms(
  data,
  scenario_source = "geco_2021",
  scenario = "1.5c",
  level = c("net", "bo_po")
)
```

Arguments

data	data.frame. Holds company-technology deviations based on PACTA for Banks TMS results. Must have been calculated according to the increasing/decreasing logic of the CA100+ calculation.
scenario_source	Character. Vector that indicates which scenario_source to use for reference in the calculation of the alignment metrics. Currently, the only supported value is "geco_2021".
scenario	Character. Vector that indicates which scenario to calculate the alignment metric for. Must be a scenario available from scenario_source.
level	Character. Vector that indicates if the aggregate alignment metric should be returned based on the net technology deviations (net) or disaggregated into build-out and phaseout technologies (bo_po).

calculate_company_tech_deviation

Return company level technology deviations for TMS sectors. To be used as input into calculation of company level aggregate alignment metrics for production trajectory sectors.

Description

Return company level technology deviations for TMS sectors. To be used as input into calculation of company level aggregate alignment metrics for production trajectory sectors.

Usage

```
calculate_company_tech_deviation(
  data,
  technology_direction,
  scenario_source = "geco_2021",
  scenario = "1.5c",
  bridge_tech = c("none", "gascap"),
  time_frame = 5L
)
```

Arguments

data	data.frame. Holds the PACTA for Banks TMS results. Must have been calculated according to the increasing/decreasing logic of the CA100+ calculation and must return unweighted company level TMSR results.
technology_direction	data frame that indicates which technologies are to be considered phase down technologies versus build out technologies

scenario_source	Character. Vector that indicates which scenario_source to use for reference in the calculation of the alignment metrics. Currently, the only supported value is "geco_2021".
scenario	Character. Vector that indicates which scenario to calculate the alignment metric for. Must be a scenario available from scenario_source.
bridge_tech	Character. Vector that indicates if a technology is considered a bridge technology. I.e. if the scenario requires a temporary build out despite the need for a long term phase down. If so, the alignment metric can be treated differently than for other technologies. Currently, the only allowed values are ("none", "gascap"). Default is "none" which means that no special calculations are applied to any technology.
time_frame	Integer of length one. The number of forward looking years that should be considered in the analysis. Standard time_frame in PACTA is five years.

create_benchmark_loanbook

Return raw loan book containing the corporate economy benchmark

Description

Return raw loan book containing the corporate economy benchmark

Usage

```
create_benchmark_loanbook(
  data,
  scenario_source,
  start_year,
  region_isos,
  benchmark_region
)
```

Arguments

data	data frame containing the asset-based company data (ABCD) in PACTA for Banks format.
scenario_source	character vector of length 1. This is used to subset the allowed regions as defined in <code>r2dii.data::region_isos</code> . It is recommended to simply use the <code>scenario_sourc_input</code> used throughout the workflow.
start_year	character vector of length 1. Defines the initial year of the analysis. The company weights will be picked based on the production capacity in the <code>start_year</code> .
region_isos	data frame containing the regional mapping for scenarios to country iso codes, following the format of <code>r2dii.data::region_isos</code>

benchmark_region

character vector of length 1. Select the region based on which the benchmark loan book should be created. Only companies with production within the selected region will be kept.

loanbook_test_data *Loan book test data*

Description

Loan book test data

Usage

```
loanbook_test_data
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 20 rows and 19 columns.

plot_sankey *Make a sankey plot*

Description

Make a sankey plot

Usage

```
plot_sankey(
  data,
  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()`

`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          Plot alignment scatterplot
```

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", "bank"),
  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 'bank' 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_animated *Plot alignment scatterplot*

Description

Plot alignment scatterplot

Usage

```
plot_scatter_animated(
  data,
  data_level = c("company", "bank"),
  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 'bank' 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,  
  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', 'group_id'.
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_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,
  matched_loanbook,
  region,
  year,
  middle_node,
  middle_node2 = NULL
)
```

Arguments

data_alignment	data.frame. Holds aggregated alignment metrics per company for tms sectors. Must contain columns: group_id, name_abcd, sector.
matched_loanbook	data.frame. Holds the matched loan books of a set of groups. Must include a column group_id and loan_size_outstanding.
region	Character. Region to filter data_alignment data frame on.
year	Integer. Year on which data_alignment should be filtered.
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	<i>Prepare data to plot scatterplot</i>
--------------	---

Description

Prepare data to plot scatterplot

Usage

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

Arguments

data_bopo	data.frame. Data containing buildout and phaseout alignment values. Must contain columns: 'group_id', 'year', 'sector', 'region', 'direction' and either 'name_abcd' and 'alignment_metric' or 'exposure_weighted_net_alignment'.
data_net	data.frame. Data containing net alignment values. Must contain columns: 'group_id', '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 'bank' 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_ids_to_plot	Character vector. Bank ids to filter on.

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("bank", "company"),  
  sector,  
  region,  
  group_ids_to_plot = NULL  
)
```

Arguments

data_bopo	data.frame. Data containing buildout and phaseout alignment values. Must contain columns: 'group_id', 'year', 'sector', 'region', 'direction' and either 'name_abcd' and 'alignment_metric' or 'exposure_weighted_net_alignment'.
data_net	data.frame. Data containing net alignment values. Must contain columns: 'group_id', '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 'bank' or 'company'.
sector	Character. Sector to filter data on.
region	Character. Region to filter data on.
group_ids_to_plot	Character vector. Group ids to filter on.

Value

data.frame

Examples

```
#TODO
```

```
prep_timeline          Prepare data to plot timeline
```

Description

Prepare data to plot timeline

Usage

```
prep_timeline(data, sector, region, group_ids_to_plot)
```

Arguments

data	data.frame. Must contain columns: 'direction', 'year', 'exposure_weighted_net_alignment', 'group_id', 'sector'.
sector	Character. Sector to filter data on.
region	Character. Region to filter data on.
group_ids_to_plot	Character vector. Group ids to filter on.

Value

data.frame

Examples

```
#TODO
```

validate_data_has_expected_cols

Validate that a data frame contains expected columns

Description

Validate that all expected columns for an operation are given in a data frame.

Usage

```
validate_data_has_expected_cols(data, expected_columns)
```

Arguments

data data frame that is to be validated

expected_columns

Character vector listing the expected columns

Index

* datasets

- abcd_test_data, [2](#)
- activity_units, [3](#)
- loanbook_test_data, [7](#)

- abcd_test_data, [2](#)
- activity_units, [3](#)
- aggregate_alignment_loanbook_exposure, [3](#)

- calculate_company_aggregate_alignment_sda, [4](#)

- calculate_company_aggregate_alignment_tms, [4](#)

- calculate_company_tech_deviation, [5](#)
- create_benchmark_loanbook, [6](#)

- loanbook_test_data, [7](#)

- plot_sankey, [7](#)
- plot_scatter, [8](#)
- plot_scatter_animated, [9](#)
- plot_timeline, [10](#)
- prep_sankey, [11](#)
- prep_scatter, [12](#)
- prep_scatter_animated, [13](#)
- prep_timeline, [14](#)

- validate_data_has_expected_cols, [15](#)