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

September 19, 2024

Title Run PACTA on multiple loan books easily

Version 0.0.0.9000

Description This repo allows running PACTA analyses on multiple loan books in a structured way and provides access to additional PACTA-related metrics for multiple loan books. Results take the form of csv files and plots and are exported to specified project paths.

Encoding UTF-8

Imports cli (>= 3.2.0), config, dplyr, ggplot2, glue, htmlwidgets, networkD3, plotly, r2dii.analysis, r2dii.data (>= 0.5.0), r2dii.match, r2dii.plot (>= 0.4.0), readr (>= 2.0.0), readxl, rlang, scales, tidyr, webshot, withr

Depends R (>= 4.1.0)

License MIT + file LICENSE

RoxygenNote 7.3.2

Suggests gt, knitr, pkgdown, rmarkdown, usethis, testthat (>= 3.1.9), tibble, writexl

Config/testthat/edition 3

Roxygen list(markdown = TRUE)

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

VignetteBuilder knitr

URL `https://rmi-pacta.github.io/pacta.multi.loanbook/`, `https://github.com/RMI-PACTA/pacta.multi.loanbook/`

LazyData true

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

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

RemoteRef HEAD

RemoteSha 63e09387e4c117833ed0c9b2d68e665acb825ff1

Contents

`abcd_test_data` *ABCD test data*

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.

`data_dictionary` *Data dictionary*

Description

An overview of the output data sets generated by the package, their data types, and the definitions of the variables.

Usage

```
data_dictionary
```

Format

`data_dictionary`:

dataset Name of the dataset
column Name of the column
typeof Data type of the column
definition Description of what the column stands for
value Which values are allowed for the column ...

Details

For more details see the help vignette: `vignette("data_dictionary", package = "pacta.multi.loanbook")`

Source

internal

`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_aggregate_loanbooks` *plot_aggregate_loanbooks*

Description

plot_aggregate_loanbooks

Usage

```
plot_aggregate_loanbooks(config)
```

Arguments

`config`	either a single string defining the path to a config YML file or a list object that contains the appropriate config params

Value

`NULL` (called for side effects)

plot_sankey *Make a sankey plot*

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
)

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

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.

`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.
`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
```

---

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", "group_var"),
  cap_outliers = NULL,
  floor_outliers = NULL
)

prep_scatter(
  data_bopo,
  data_net,
  data_level = c("group_var", "company"),
  year,
  sector,
  region,
  group_var,
  groups_to_plot = 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 to filter data on.

`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 on which the data should be filtered.

`region` Character. Region to filter data on.

`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.

`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'`.

`group_var` Character. Vector of length 1. Variable to group by.

`groups_to_plot` Character vector. Groups to filter on.

Value

object of type "ggplot"

data.frame

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
)

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

Arguments

`data` data.frame. Holds net aggregated alignment metrics on the loan book level. Must contain columns: `"scenario"`, `"region"`, `"sector"`, `"year"`, `"exposure_weighted_net_alignme` `"sum_loan_size_outstanding"` 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. Vector of length 1. A column to group by. Must be available variables in `data`.

`currency` Character. Currency to display in the plot labels.

`year` Integer. Year on which `data` should be filtered.

`region` Character. Region to filter `data` data frame on.

`scenario` Character. Scenario to filter `data` data frame on.

`exclude_groups` Character. Character specifying any values from `group_var` 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

object of type "ggplot"

data.frame

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
)

prep_scatter_animated(
  data_bopo,
  data_net,
  data_level = c("group_var", "company"),
  sector,
  region,
  group_var,
  groups_to_plot = 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 to filter data on.

`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 filter data on.

`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.

`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'`.

`group_var` Character. Vector of length 1. Variable to group by.

`groups_to_plot` Character vector. Groups to filter on.

Value

object of type "plotly"

data.frame

Examples

```
# TODO
```

run_pacta *run_pacta*

Description

run_pacta

Usage

```
run_pacta(config)
```

Arguments

`config` either a single string defining the path to a config YML file or a list object that contains the appropriate config params

Value

`NULL` (called for side effects)

Index