

Package ‘r2dii.plot’

July 23, 2025

Title Visualize the Climate Scenario Alignment of a Financial Portfolio

Version 0.5.2

Description Create plots to visualize the alignment of a corporate lending financial portfolio to climate change scenarios based on climate indicators (production and emission intensities) across key climate relevant sectors of the 'PACTA' methodology (Paris Agreement Capital Transition Assessment; <<https://www.transitionmonitor.com/>>). Financial institutions use 'PACTA' to study how their capital allocation decisions align with climate change mitigation goals.

License MIT + file LICENSE

URL <https://rmi-pacta.github.io/r2dii.plot/>,
<https://github.com/RMI-PACTA/r2dii.plot>

BugReports <https://github.com/RMI-PACTA/r2dii.plot/issues>

Depends R (>= 3.5)

Imports dplyr, ggplot2, ggrepel, glue, r2dii.data, rlang, stringr, scales

Suggests cli, covr, r2dii.analysis, r2dii.match, readr, rmarkdown, spelling, testthat (>= 3.0.0), waldo

Config/testthat/edition 3

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

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.3.2

NeedsCompilation no

Author Monika Furdyna [aut, ctr, cre] (ORCID:
<<https://orcid.org/0000-0002-3728-0646>>),
Mauro Lepore [aut, ctr] (ORCID:
<<https://orcid.org/0000-0002-1986-7988>>),

Alex Axthelm [aut, ctr] (ORCID:
<https://orcid.org/0000-0001-8579-8565>),
 RMI [cph, fnd]

Maintainer Monika Furdyna <monika.furdyna@gmail.com>

Repository CRAN

Date/Publication 2025-06-21 13:30:02 UTC

Contents

| | |
|-------------------------------------|-----------|
| data_dictionary | 2 |
| market_share_demo | 3 |
| plot_emission_intensity | 4 |
| plot_techmix | 4 |
| plot_trajectory | 5 |
| prep_emission_intensity | 6 |
| prep_techmix | 7 |
| prep_trajectory | 8 |
| qplot_emission_intensity | 9 |
| qplot_techmix | 10 |
| qplot_trajectory | 11 |
| r2dii_colours | 12 |
| scale_colour_r2dii | 13 |
| scale_colour_r2dii_sector | 14 |
| scale_colour_r2dii_tech | 15 |
| sda_demo | 16 |
| theme_2dii | 16 |
| to_title | 17 |
| Index | 19 |

| | |
|-----------------|------------------------|
| data_dictionary | <i>Data Dictionary</i> |
|-----------------|------------------------|

Description

A table of column names and descriptions of data frames used or exported by the functions in this package.

Usage

data_dictionary

Format

data_dictionary:

dataset Name of the dataset

column Name of the column

typeof Type of the column

definition Definition of the column

Examples

data_dictionary

| | |
|-------------------|---|
| market_share_demo | <i>An example of a market_share_demo-like dataset</i> |
|-------------------|---|

Description

Dataset imitating the output of `r2dii.analysis::target_market_share()`.

Usage

market_share_demo

Format

An object of class `spec_tbl_df` (inherits from `tbl_df`, `tbl`, `data.frame`) with 802 rows and 10 columns.

See Also

`r2dii.analysis::target_market_share()`.

Other datasets: `r2dii_colours`, `sda_demo`

Examples

market_share_demo

plot_emission_intensity
Create an emission intensity plot

Description

Create an emission intensity plot

Usage

```
plot_emission_intensity(data)
```

Arguments

data A data frame like the output of prep_emission_intensity().

Value

An object of class "ggplot".

See Also

[sda_demo](#).

Examples

```
# plot with `qplot_emission_intensity()` parameters
data <- subset(sda_demo, sector == "cement" & region == "global")
data <- prep_emission_intensity(data, span_5yr = TRUE, convert_label = to_title)

plot_emission_intensity(data)
```

plot_techmix *Create a techmix plot*

Description

Create a techmix plot

Usage

```
plot_techmix(data)
```

Arguments

data A data frame like the output of prep_techmix().

Value

An object of class "ggplot".

See Also

[market_share_demo](#).

Examples

```
# plot with `qplot_techmix()` parameters
data <- subset(
  market_share_demo,
  scenario_source == "demo_2020" &
  sector == "power" &
  region == "global" &
  metric %in% c("projected", "corporate_economy", "target_sds")
)
data <- prep_techmix(
  data,
  span_5yr = TRUE,
  convert_label = recode_metric_techmix,
  convert_tech_label = spell_out_technology
)

plot_techmix(data)
```

plot_trajectory *Create a trajectory plot*

Description

Create a trajectory plot

Usage

```
plot_trajectory(data, center_y = FALSE, perc_y_scale = FALSE)
```

Arguments

| | |
|--------------|---|
| data | A data frame like the outputs of <code>prep_trajectory()</code> . <ul style="list-style-type: none"> (Optional) If present, the column label is used for data labels. |
| center_y | Logical. Use TRUE to center the y-axis around start value (the default behavior of <code>qplot_trajectory()</code>), or use FALSE to not center. |
| perc_y_scale | Logical. FALSE defaults to using no label conversion. Use TRUE to convert labels on y-axis to percentage using <code>scales::percent</code> (the default behavior of <code>qplot_trajectory()</code>). |

Value

An object of class "ggplot".

See Also

[market_share_demo](#).

Examples

```
# plot with `qplot_trajectory()` parameters
data <- subset(
  market_share_demo,
  sector == "power" &
  technology == "renewables" &
  region == "global" &
  scenario_source == "demo_2020"
)
data <- prep_trajectory(data)

plot_trajectory(
  data,
  center_y = TRUE,
  perc_y_scale = TRUE
)
```

prep_emission_intensity

Prepare data for a emission intensity plot

Description

Prepare data for a emission intensity plot

Usage

```
prep_emission_intensity(data, convert_label = identity, span_5yr = FALSE)
```

Arguments

- | | |
|---------------|---|
| data | A data frame. Requirements: <ul style="list-style-type: none">• The structure must be like sda_demo.• The column sector must have a single value (e.g. "cement").• (Optional) If present, the column label is used for data labels. |
| convert_label | A symbol. The unquoted name of a function to apply to y-axis labels. For example: <ul style="list-style-type: none">• To convert labels to uppercase use convert_label = toupper. |

- To get the default behavior of `qplot_emission_intensity` use `convert_label = to_title`.
- `span_5yr` Logical. Use TRUE to restrict the time span to 5 years from the start year (the default behavior of `qplot_emission_intensity()`), or use FALSE to impose no restriction.

Value

A data-frame ready to be plotted using `plot_emission_intensity()`.

See Also

[sda_demo](#).

Examples

```
# `data` must meet documented "Requirements"
data <- subset(sda_demo, sector == "cement" & region == "global")
prep_emission_intensity(data)
```

```
prep_techmix
```

Prepare data for plotting technology mix

Description

Prepare data for plotting technology mix

Usage

```
prep_techmix(
  data,
  convert_label = identity,
  span_5yr = FALSE,
  convert_tech_label = identity
)
```

Arguments

- `data` A data frame. Requirements:
- The structure must be like [market_share_demo](#).
 - The following columns must have a single value: `sector`, `region`, `scenario_source`.
 - The column `metric` must have a portfolio (e.g. "projected"), a benchmark (e.g. "corporate_economy"), and a single scenario (e.g. "target_sds").
 - (Optional) If present, the column `label` is used for data labels.
 - (Optional) If present, the column `label_tech` is used for technology labels.
- `convert_label` A symbol. The unquoted name of a function to apply to y-axis labels. For example:

- To convert labels to uppercase use `convert_label = toupper`.
 - To get the default behavior of `qplot_techmix` use `convert_label = recode_metric_techmix`.
- `span_5yr` Logical. Use TRUE to restrict the time span to 5 years from the start year (the default behavior of `qplot_techmix()`), or use FALSE to impose no restriction.
- `convert_tech_label` A symbol. The unquoted name of a function to apply to technology legend labels. For example, to convert labels to uppercase use `convert_tech_label = toupper`. To get the default behavior of `qplot_techmix()` use `convert_tech_label = spell_out_technology`.

Value

A data-frame ready to be plotted using `plot_techmix()`.

See Also

[market_share_demo](#).

Examples

```
# `data` must meet documented "Requirements"
data <- subset(
  market_share_demo,
  scenario_source == "demo_2020" &
  sector == "power" &
  region == "global" &
  metric %in% c("projected", "corporate_economy", "target_sds")
)

prep_techmix(data)
```

```
prep_trajectory
```

```
Prepare data for a trajectory plot
```

Description

Prepare data for a trajectory plot

Usage

```
prep_trajectory(
  data,
  convert_label = identity,
  span_5yr = FALSE,
  value_col = "percentage_of_initial_production_by_scope"
)
```


Arguments

| | |
|---------------|---|
| data | A data frame. Requirements: <ul style="list-style-type: none"> • The structure must be like market_share_demo. • The following columns must have a single value: sector, technology, region, scenario_source. • (Optional) If present, the column label is used for data labels. |
| convert_label | A symbol. The unquoted name of a function to apply to y-axis labels. For example: <ul style="list-style-type: none"> • To convert labels to uppercase use <code>convert_label = toupper</code>. • To get the default behavior of <code>qplot_trajectory</code> use <code>convert_label = recode_metric_trajectory</code>. |
| span_5yr | Logical. Use TRUE to restrict the time span to 5 years from the start year (the default behavior of <code>qplot_trajectory()</code>), or use FALSE to impose no restriction. |
| value_col | Character. Name of the column to be used as a value to be plotted. |

Value

A data-frame ready to be plotted using `plot_trajectory()`.

See Also

[market_share_demo](#).

Examples

```
# `data` must meet documented "Requirements"
data <- subset(
  market_share_demo,
  sector == "power" &
  technology == "renewables" &
  region == "global" &
  scenario_source == "demo_2020"
)

prep_trajectory(data)
```

qplot_emission_intensity

Create a quick emission intensity plot

Description

Compared to `plot_emission_intensity()` this function:

- is restricted to plotting future as 5 years from the start year,
- outputs formatted labels, based on emission metric column,
- outputs a title,
- outputs formatted axis labels.

Usage

```
qplot_emission_intensity(data)
```

Arguments

data A data frame like the output of `prep_emission_intensity()`.

Value

An object of class "ggplot".

See Also

`plot_emission_intensity`

Examples

```
# `data` must meet documented "Requirements"
data <- subset(sda_demo, sector == "cement" & region == "global")

qplot_emission_intensity(data)
```

| | |
|---------------|------------------------------------|
| qplot_techmix | <i>Create a quick techmix plot</i> |
|---------------|------------------------------------|

Description

Compared to `plot_techmix()` this function:

- is restricted to plotting future as 5 years from the start year,
- outputs pretty bar labels, based on metric column,
- outputs pretty legend labels, based on technology column,
- outputs a title.

Usage

```
qplot_techmix(data)
```

Arguments

data A data frame like the output of `prep_techmix()`.

Value

An object of class "ggplot".

See Also

plot_techmix

Examples

```
# `data` must meet documented "Requirements"
data <- subset(
  market_share_demo,
  sector == "power" &
  region == "global" &
  scenario_source == "demo_2020" &
  metric %in% c("projected", "corporate_economy", "target_sds")
)

qplot_techmix(data)
```

qplot_trajectory *Create a quick trajectory plot*

Description

Compared to [plot_trajectory\(\)](#) this function:

- is restricted to plotting only 5 years from the start year,
- outputs pretty legend labels, based on the column holding metrics,
- outputs a title,
- outputs a subtitle,
- outputs informative axis labels in sentence case.

Usage

```
qplot_trajectory(data)
```

Arguments

`data` A data frame like the outputs of [prep_trajectory\(\)](#).

- (Optional) If present, the column label is used for data labels.

Value

An object of class "ggplot".

See Also

plot_trajectory

Examples

```
# `data` must meet documented "Requirements"
data <- subset(
  market_share_demo,
  sector == "power" &
  technology == "renewablescap" &
  region == "global" &
  scenario_source == "demo_2020"
)

qplot_trajectory(data)
```

r2dii_colours

Colour datasets

Description

All datasets have at least two columns:

- label: Text label of the colour.
- hex: Hex code of the colour.

Usage

palette_colours

scenario_colours

sector_colours

technology_colours

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 9 rows and 2 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 5 rows and 2 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 8 rows and 2 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 18 rows and 3 columns.

Details

In `scenario_colours`, colours are ordered from red to green to be used in trajectory charts.

See Also

Other datasets: [market_share_demo](#), [sda_demo](#)

Examples

```
palette_colours
scenario_colours
sector_colours
technology_colours
```

scale_colour_r2dii *Custom PACTA colour and fill scales*

Description

A custom discrete colour and fill scales with colours from the PACTA palette.

Usage

```
scale_colour_r2dii(colour_labels = NULL, ...)
scale_fill_r2dii(colour_labels = NULL, ...)
```

Arguments

`colour_labels` A character vector. Specifies colour labels to use and their order. Run `unique(r2dii.plot::palette_colours)` to see available colours. Similar to value parameter in `ggplot2::scale_colour_manual()`.

`...` Other parameters passed on to `ggplot2::discrete_scale()`.

Value

An object of class "ScaleDiscrete".

See Also

Other r2dii scales: `scale_colour_r2dii_sector()`, `scale_colour_r2dii_tech()`

Examples

```
library(ggplot2, warn.conflicts = FALSE)

ggplot(mpg) +
  geom_point(aes(displ, hwy, color = class)) +
  scale_colour_r2dii()

ggplot(mpg) +
  geom_histogram(aes(cyl, fill = class), position = "dodge", bins = 5) +
  scale_fill_r2dii()
```

`scale_colour_r2dii_sector`*Custom PACTA sector colour and fill scales*

Description

A custom discrete colour and fill scales with colours from the PACTA sector palette.

Usage

```
scale_colour_r2dii_sector(sectors = NULL, ...)
```

```
scale_fill_r2dii_sector(sectors = NULL, ...)
```

Arguments

`sectors` A character vector. Specifies sector colours to use and their order. Run `unique(r2dii.plot:::sector_c)` to see available labels. Similar to `value` parameter in `ggplot2::scale_colour_manual()`.

`...` Other parameters passed on to `ggplot2::discrete_scale()`.

Value

An object of class "ScaleDiscrete".

See Also

Other r2dii scales: `scale_colour_r2dii()`, `scale_colour_r2dii_tech()`

Examples

```
library(ggplot2, warn.conflicts = FALSE)
```

```
ggplot(mpg) +  
  geom_point(aes(displ, hwy, color = class)) +  
  scale_colour_r2dii_sector()
```

```
ggplot(mpg) +  
  geom_histogram(aes(cyl, fill = class), position = "dodge", bins = 5) +  
  scale_fill_r2dii_sector()
```

`scale_colour_r2dii_tech`*Custom PACTA technology colour and fill scales*

Description

A custom discrete colour and fill scales with colours from the PACTA technology palette.

Usage

```
scale_colour_r2dii_tech(sector, technologies = NULL, ...)
```

```
scale_fill_r2dii_tech(sector, technologies = NULL, ...)
```

Arguments

| | |
|---------------------------|---|
| <code>sector</code> | A string. Sector name specifying a colour palette. Run <code>unique(r2dii.plot::technology_colours\$sector)</code> to see available sectors. |
| <code>technologies</code> | A character vector. Specifies technologies to use as colours and their order. Run <code>unique(r2dii.plot::technology_colours\$technology)</code> to see available technologies (pay attention if they match the sector). Similar to value parameter in <code>ggplot2::scale_colour_manual()</code> . |
| <code>...</code> | Other parameters passed on to <code>ggplot2::discrete_scale()</code> . |

Value

An object of class "ScaleDiscrete".

See Also

Other r2dii scales: [scale_colour_r2dii\(\)](#), [scale_colour_r2dii_sector\(\)](#)

Examples

```
library(ggplot2, warn.conflicts = FALSE)

ggplot(mpg) +
  geom_point(aes(displ, hwy, color = class)) +
  scale_colour_r2dii_tech("automotive")

ggplot(mpg) +
  geom_histogram(aes(cyl, fill = class), position = "dodge", bins = 5) +
  scale_fill_r2dii_tech("automotive")
```

`sda_demo`*An example of an sda_demo-like dataset*

Description

Dataset imitating the output of `r2dii.analysis::target_sda()`.

Usage

```
sda_demo
```

Format

An object of class `spec_tbl_df` (inherits from `tbl_df`, `tbl`, `data.frame`) with 110 rows and 6 columns.

Source

<https://github.com/RMI-PACTA/r2dii.plot/issues/55>.

See Also

`r2dii.analysis::target_sda()`.

Other datasets: `market_share_demo`, `r2dii_colours`

Examples

```
sda_demo
```

`theme_2dii`*Complete theme*

Description

A ggplot theme which can be applied to all graphs to appear according to the PACTA plotting aesthetics.

Usage

```
theme_2dii(  
  base_size = 12,  
  base_family = "Helvetica",  
  base_line_size = base_size/22,  
  base_rect_size = base_size/22  
)
```


Arguments

base_size base font size, given in pts.
 base_family base font family
 base_line_size base size for line elements
 base_rect_size base size for rect elements

Value

An object of class "theme", "gg".

See Also

[ggplot2::theme_classic](#).

Examples

```
library(ggplot2, warn.conflicts = FALSE)

ggplot(mtcars) +
  geom_histogram(aes(mpg), bins = 10) +
  theme_2dii()
```

 to_title

Replicate labels produced with qplot_() functions*

Description

- to_title() converts labels like [qplot_emission_intensity\(\)](#).
- recode_metric_trajectory() converts labels like [qplot_trajectory\(\)](#).
- recode_metric_techmix() converts labels like [qplot_techmix\(\)](#).
- spell_out_technology() converts technology labels like [qplot_techmix\(\)](#).

Usage

```
to_title(x)

recode_metric_techmix(x)

recode_metric_trajectory(x)

spell_out_technology(x)
```

Arguments

x A character vector.

Value

A character vector.

Examples

```
to_title(c("a.string", "another_STRING"))
```

```
metric <- c("projected", "corporate_economy", "target_xyz", "else")  
recode_metric_trajectory(metric)
```

```
recode_metric_techmix(metric)
```

```
spell_out_technology(c("gas", "ice", "coalcap", "hdv"))
```

Index

- * **data dictionary**
 - data_dictionary, 2
- * **datasets**
 - data_dictionary, 2
 - market_share_demo, 3
 - r2dii_colours, 12
 - sda_demo, 16
- * **plotting functions**
 - theme_2dii, 16
- * **r2dii scales**
 - scale_colour_r2dii, 13
 - scale_colour_r2dii_sector, 14
 - scale_colour_r2dii_tech, 15
- data_dictionary, 2
- ggplot2::discrete_scale(), 13–15
- ggplot2::scale_colour_manual(), 13–15
- ggplot2::theme_classic, 17
- market_share_demo, 3, 5–9, 12, 16
- palette_colours (r2dii_colours), 12
- plot_emission_intensity, 4
- plot_emission_intensity(), 9
- plot_techmix, 4
- plot_techmix(), 10
- plot_trajectory, 5
- plot_trajectory(), 11
- prep_emission_intensity, 6
- prep_techmix, 7
- prep_trajectory, 8
- qplot_emission_intensity, 9
- qplot_emission_intensity(), 17
- qplot_techmix, 10
- qplot_techmix(), 17
- qplot_trajectory, 11
- qplot_trajectory(), 17
- r2dii.analysis::target_market_share(), 3
- r2dii.analysis::target_sda(), 16
- r2dii_colours, 3, 12, 16
- recode_metric_techmix (to_title), 17
- recode_metric_trajectory (to_title), 17
- scale_color_r2dii (scale_colour_r2dii), 13
- scale_color_r2dii_sector (scale_colour_r2dii_sector), 14
- scale_color_r2dii_tech (scale_colour_r2dii_tech), 15
- scale_colour_r2dii, 13, 14, 15
- scale_colour_r2dii_sector, 13, 14, 15
- scale_colour_r2dii_tech, 13, 14, 15
- scale_fill_r2dii (scale_colour_r2dii), 13
- scale_fill_r2dii_sector (scale_colour_r2dii_sector), 14
- scale_fill_r2dii_tech (scale_colour_r2dii_tech), 15
- scenario_colours (r2dii_colours), 12
- sda_demo, 3, 4, 6, 7, 12, 16
- sector_colours (r2dii_colours), 12
- spell_out_technology (to_title), 17
- technology_colours (r2dii_colours), 12
- theme_2dii, 16
- to_title, 17