Package ‘r2dii.plot’

February 29, 2024

Title  Visualize the Climate Scenario Alignment of a Financial Portfolio

Version 0.4.0

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

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

Depends  R (>= 3.4)

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

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

Config/testthat/edition  3

Encoding  UTF-8

Language  en-US

LazyData  true

RoxygenNote  7.3.1

NeedsCompilation  no

Author  Monika Furdyina [aut, ctr, cre]

     (<https://orcid.org/0000-0002-3728-0646>),
     Mauro Lepore [aut, ctr] (<https://orcid.org/0000-0002-1986-7988>),
     Alex Axthelm [aut, ctr] (<https://orcid.org/0000-0001-8579-8565>),
     Rocky Mountain Institute [cph, fnd]
market_share

Description

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

Usage

market_share

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`
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.

Examples
# plot with `qplot_emission_intensity()`' parameters
data <- subset(sda, sector == "cement" & region == "global") %>%
    prep_emission_intensity(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.

Examples

# plot with `qplot_techmix()` parameters
data <- subset(
market_share,
scenario_source == "demo_2020" &
sector == "power" &
region == "global" &
metric %in% c("projected", "corporate_economy", "target_sds")
)

prep_techmix(
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 prep_trajectory().
• (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 qplot_trajectory()), 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 scales::percent (the default behavior of qplot_trajectory()).

Value
An object of class "ggplot".

See Also
market_share.

Examples
# plot with `qplot_trajectory()` parameters
data <- subset(
  market_share,
  sector == "power" &
  technology == "renewablescape" &
  region == "global" &
  scenario_source == "demo_2020"
) %>%
  prep_trajectory()

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:
  • The structure must be like sda.
  • 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:
  • 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.

Examples

# 'data' must meet documented "Requirements"
data <- subset(sda, 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.
  • 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.
Examples

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

prep_technix(data)
```

Description

Prepare data for a trajectory plot

Usage

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

Arguments

- **data**: A data frame. Requirements:
  - The structure must be like `market_share`.
  - 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:
  - To convert labels to uppercase use `convert_label = toupper`.
  - To get the default behavior of ``

- **span_5yr**: Logical. Use `TRUE` to restrict the time span to 5 years from the start year (the default behavior of `qplot_trajectory()`), 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()`.
qplot_emission_intensity

See Also

market_share.

Examples

```r
# `data` must meet documented "Requirements"
data <- subset(
  market_share,
  sector == "power" &
  technology == "renewables cap" &
  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

```r
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, sector == "cement" & region == "global")

qplot_emission_intensity(data)

qplot_techmix(data)  

Create a quick techmix plot

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,
  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**

```r
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**

```r
# 'data' must meet documented "Requirements"
data <- subset(
  market_share,
  sector == "power" &
  technology == "renewables_cap" &
  region == "global" &
  scenario_source == "demo_2020"
)
qplot_trajectory(data)
```
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`, `sda`

Examples

- `palette_colours`
- `scenario_colours`
- `sector_colours`
- `technology_colours`
scale_colour_r2dii

Custom 2DII colour and fill scales

Description

A custom discrete colour and fill scales with colours from 2DII 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$label) 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 2DII sector colour and fill scales*

**Description**

A custom discrete colour and fill scales with colours from 2DII sector palette.

**Usage**

```r
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_colours$label)` 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**

```r
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 2DII technology colour and fill scales

Description
A custom discrete colour and fill scales with colours from 2DII technology palette.

Usage
scale_colour_r2dii_tech(sector, technologies = NULL, ...)
scale_fill_r2dii_tech(sector, technologies = NULL, ...)

Arguments
sector          A string. Sector name specifying a colour palette. Run unique(r2dii.plot:::technology_colours$sector) to see available sectors.
technologies    A character vector. Specifies technologies to use as colours and their order. Run unique(r2dii.plot:::technology_colours$technology) to see available technologies (pay attention if they match the sector). 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_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  

An example of an sda-like dataset

Description

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

Usage

sda

Format

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

Source


See Also

`r2dii.analysis::target_sda()`.

Other datasets: `market_share`, `r2dii_colours`

Examples

sda

theme_2dii  

Complete theme

Description

A ggplot theme which can be applied to all graphs to appear according to 2DII plotting aesthetics.

Usage

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

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

* datasets
  market_share, 2
  r2dii_colours, 12
  sda, 16

* plotting functions
  theme_2dii, 16

* r2dii scales
  scale_colour_r2dii, 13
  scale_colour_r2dii_sector, 14
  scale_colour_r2dii_tech, 15

  ggplot2::discrete_scale(), 13–15
  ggplot2::scale_colour_manual(), 13–15
  ggplot2::theme_classic, 17

  market_share, 2, 4, 5, 7–9, 12, 16

  palette_colours (r2dii_colours), 12
  plot_emission_intensity, 3
  plot_emission_intensity(), 9
  plot_technix, 4
  plot_technix(), 10
  plot_trajectory, 5
  plot_trajectory(), 11
  prep_emission_intensity, 6
  prep_technix, 7
  prep_trajectory, 8

  qplot_emission_intensity, 9
  qplot_emission_intensity(), 17
  qplot_technix, 10
  qplot_technix(), 17
  qplot_trajectory, 11
  qplot_trajectory(), 17

  r2dii.analysis::target_market_share(), 2
  r2dii.analysis::target_sda(), 16
  r2dii_colours, 2, 12, 16
  recode_metric_technix (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, 2, 3, 6, 12, 16
  sector_colours (r2dii_colours), 12
  spell_out_technology (to_title), 17

  technology_colours (r2dii_colours), 12
  theme_2dii, 16
  to_title, 17