Package 'r2dii.plot'

October 14, 2022

Title Visualize the Climate Scenario Alignment of a Financial Portfolio

Version 0.3.0

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://2degrees-investing.org/>). 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/2DegreesInvesting/r2dii.plot, https://2degreesinvesting.github.io/r2dii.plot/

BugReports https://github.com/2DegreesInvesting/r2dii.plot/issues

Depends R (>= 3.4)

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

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 7.1.2

NeedsCompilation no

Author Monika Furdyna [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>), 2 Degrees Investing Initiative [cph, fnd]

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

Repository CRAN

Date/Publication 2022-05-05 23:20:10 UTC

R topics documented:

`market_share` *An example of a* `market_share`*-like dataset*

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: `sda`

Examples

```
market_share
```

`plot_emission_intensity` *Create an emission intensity plot*

Description

Create an emission intensity plot

Usage

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

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.
`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.
`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`.

Value

An object of class "ggplot".

See Also

sda.

Examples

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

# plot with `qplot_emission_intensity()` parameters
plot_emission_intensity(
  data,
  span_5yr = TRUE,
  convert_label = to_title
)
```

`plot_techmix` *Create a techmix plot*

Description

Create a techmix plot

Usage

```
plot_techmix(
  data,
  span_5yr = FALSE,
  convert_label = identity,
  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.
`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_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`.
`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

An object of class "ggplot".

See Also

market_share.

Examples

```
# `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")
)

plot_techmix(data)

# plot with `qplot_techmix()` parameters
plot_techmix(
  data,
  span_5yr = TRUE,
  convert_label = recode_metric_techmix,
  convert_tech_label = spell_out_technology
)
```

`plot_trajectory` *Create a trajectory plot*

Description

Create a trajectory plot

Usage

```
plot_trajectory(
  data,
  span_5yr = FALSE,
  convert_label = identity,
  center_y = FALSE,
  value_col = "percentage_of_initial_production_by_scope",
  perc_y_scale = FALSE
)
```

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.
`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.

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_trajectory()` use `convert_label = recode_metric_trajectory`.

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.

value_col
: Character. Name of the column to be used as a value to be plotted.

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

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

plot_trajectory(data)

# plot with `qplot_trajectory()` parameters
plot_trajectory(
  data,
  span_5yr = TRUE,
  convert_label = recode_metric_trajectory,
  center_y = TRUE,
  value_col = "percentage_of_initial_production_by_scope",
  perc_y_scale = TRUE
)
```

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. 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.

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 *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. 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.

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

```
qplot_trajectory(data)
```

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.

Value

An object of class "ggplot".

See Also

plot_trajectory

Examples

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

qplot_trajectory(data)
```

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(labels = NULL, ...)

scale_fill_r2dii(labels = NULL, ...)
```

Arguments

labels	A character vector. Specifies colour labels to use and their order. Run `unique(r2dii.plot:::palette_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_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

```
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_tech(), scale_colour_r2dii()

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 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$se` to see available sectors.
technologies	A character vector. Specifies technologies to use as colours and their order. Run `unique(r2dii.plot:::technology_colours$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_sector()`, `scale_colour_r2dii()`

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

https://github.com/2DegreesInvesting/r2dii.plot/issues/55.

See Also

`r2dii.analysis::target_sda()`.

Other datasets: `market_share`

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

```
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