Package: medacoPlot (via r-universe)

November 19, 2024

Title Renders plots of medaco csv data

Version 0.1.1

Description Plot power input and output.

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports dplyr (>= 1.1.4), ggplot2 (>= 3.5.1), ggridges (>= 0.5.6), magrittr, readr (>= 2.1.5), rlang, shiny (>= 1.9.1), shinyFiles (>= 0.9.3), tibble, tidyr (>= 1.3.1)

Suggests testthat (>= 3.0.0), vdiffr (>= 1.0.7)

Config/testthat/edition 3

License MIT + file LICENSE

URL https://joergsesterhenn.github.io/medaco-plot/

Repository https://joergsesterhenn.r-universe.dev

RemoteUrl https://github.com/joergsesterhenn/medaco-plot

RemoteRef HEAD

RemoteSha d2368db336fe46dc168b2dfd5bc2313f776d496f

Contents

`get_files_in_path` *Get List of Files in Directory*

Description

Retrieves a list of all `.CSV` files in the specified directory.

Usage

```
get_files_in_path(data_path)
```

Arguments

`data_path`	Character string specifying the directory path to search.

Value

A character vector of file paths for `.CSV` files in the directory.

Examples

```
get_files_in_path("data/")
```

`get_hourly_data_long` *Get Hourly Data in Long Format*

Description

Aggregates and reshapes the data by hour, returning it in a long format.

Usage

```
get_hourly_data_long(power_data)
```

Arguments

`power_data`	data frame with `timestamp`, `INPUT`, and `OUTPUT` columns.

Value

A data frame with hourly `total_input` and `total_output` values.

Examples

```
# Example using a small sample data frame
power_data <- data.frame(
  timestamp = c(
       as.POSIXct("2000-01-01 01:00:00", tz = "UTC"),
       as.POSIXct("2000-01-02 01:00:00", tz = "UTC"),
       as.POSIXct("2000-02-01 02:00:00", tz = "UTC"),
       as.POSIXct("2000-02-02 02:00:00", tz = "UTC")),
  INPUT = c(1.0, 2.0, 3.0, 4.0),
  OUTPUT = c(4.0, 3.0, 2.0, 1.0)
)
get_hourly_data_long(power_data)
```

---

get_hourly_monthly_data_long

Get Hourly and Monthly Data in Long Format

Description

Aggregates and reshapes the data by both hour and month, returning it.

Usage

```
get_hourly_monthly_data_long(power_data)
```

Arguments

`power_data`	data frame with `timestamp`, `INPUT`, and `OUTPUT` columns.

Value

A data frame with hourly and monthly values.

Examples

```
# Example using a small sample data frame
power_data <- data.frame(
  timestamp = c(
       as.POSIXct("2000-01-01 01:00:00", tz = "UTC"),
       as.POSIXct("2000-01-02 01:00:00", tz = "UTC"),
       as.POSIXct("2000-02-01 02:00:00", tz = "UTC"),
       as.POSIXct("2000-02-02 02:00:00", tz = "UTC")),
  INPUT = c(1.0, 2.0, 3.0, 4.0),
  OUTPUT = c(4.0, 3.0, 2.0, 1.0)
)
get_hourly_monthly_data_long(power_data)
```

get_monthly_data_long *Get Monthly Data in Long Format*

Description

Aggregates and reshapes the data by month, returning it in a long format.

Usage

```
get_monthly_data_long(power_data)
```

Arguments

power_data	data frame with timestamp, INPUT, and OUTPUT columns.

Value

A data frame with monthly total_input and total_output values.

Examples

```
# Example using a small sample data frame
power_data <- data.frame(
  timestamp = c(
       as.POSIXct("2000-01-01 01:00:00", tz = "UTC"),
       as.POSIXct("2000-01-02 01:00:00", tz = "UTC"),
       as.POSIXct("2000-02-01 02:00:00", tz = "UTC"),
       as.POSIXct("2000-02-02 02:00:00", tz = "UTC")),
  INPUT = c(1.0, 2.0, 3.0, 4.0),
  OUTPUT = c(4.0, 3.0, 2.0, 1.0)
)
get_monthly_data_long(power_data)
```

get_yearly_data_long *Get Monthly Data in Long Format*

Description

Aggregates and reshapes the data by year, returning it in a long format.

Usage

```
get_yearly_data_long(power_data)
```

Arguments

power_data	data frame with timestamp, INPUT, and OUTPUT columns.

Value

A data frame with yearly `total_input` and `total_output` values.

Examples

```
# Example using a small sample data frame
power_data <- data.frame(
  timestamp = c(
       as.POSIXct("2000-01-01 01:00:00", tz = "UTC"),
       as.POSIXct("2000-01-02 01:00:00", tz = "UTC"),
       as.POSIXct("2000-02-01 02:00:00", tz = "UTC"),
       as.POSIXct("2000-02-02 02:00:00", tz = "UTC")),
  INPUT = c(1.0, 2.0, 3.0, 4.0),
  OUTPUT = c(4.0, 3.0, 2.0, 1.0)
)
get_monthly_data_long(power_data)
```

---

pivot_longer_data *Pivot Data to Long Format for Input and Output*

Description

Transforms data from a wide format to a long format, creating separate rows for `total_input` and `total_output`.

Usage

```
pivot_longer_data(power_data)
```

Arguments

`power_data`	data frame with `timestamp`, `INPUT`, and `OUTPUT` columns.

Value

A data frame in long format with a `type` column indicating input or output, and a `value` column for the corresponding values.

Examples

```
# Example using a small sample data frame
power_data <- data.frame(
  timestamp = c(
       as.POSIXct("2000-01-01 01:00:00", tz = "UTC"),
       as.POSIXct("2000-01-02 01:00:00", tz = "UTC"),
       as.POSIXct("2000-02-01 02:00:00", tz = "UTC"),
       as.POSIXct("2000-02-02 02:00:00", tz = "UTC")),
  total_input = c(1.0, 2.0, 3.0, 4.0),
  total_output = c(4.0, 3.0, 2.0, 1.0)
```

```
)
pivot_longer_data(power_data)
```

`plot` *Generic Plot Function*

Description

Selects a plot function based on the specified type and plots the dataset.

Usage

```
plot(plot_type, power_data)
```

Arguments

`plot_type`	Character, the plot type from the dropdown menu.
`power_data`	data frame with `timestamp`, `INPUT`, and `OUTPUT` columns.

Value

A ggplot object created by the appropriate plotting function.

Examples

```
# Example using a small sample data frame
power_data <- data.frame(
  timestamp = c(
    as.POSIXct("2000-01-01 01:00:00", tz = "UTC"),
    as.POSIXct("2000-01-02 01:00:00", tz = "UTC"),
    as.POSIXct("2000-02-01 02:00:00", tz = "UTC"),
    as.POSIXct("2000-02-02 02:00:00", tz = "UTC")
  ),
  INPUT = c(1.0, 2.0, 3.0, 4.0),
  OUTPUT = c(4.0, 3.0, 2.0, 1.0)
)
plot("by month", power_data)
```

plot_aggregated_by_hour *Plot Aggregated Data by Hour*

Description

Generates a bar plot showing hourly input and output sums.

Usage

```
plot_aggregated_by_hour(power_data)
```

Arguments

power_data	data frame with timestamp, INPUT, and OUTPUT columns.

Value

A ggplot object showing hourly aggregated values.

Examples

```
# Example using a small sample data frame
power_data <- data.frame(
  timestamp = c(
    as.POSIXct("2000-01-01 01:00:00", tz = "UTC"),
    as.POSIXct("2000-01-02 01:00:00", tz = "UTC"),
    as.POSIXct("2000-02-01 02:00:00", tz = "UTC"),
    as.POSIXct("2000-02-02 02:00:00", tz = "UTC")
  ),
  INPUT = c(1.0, 2.0, 3.0, 4.0),
  OUTPUT = c(4.0, 3.0, 2.0, 1.0)
)
plot_aggregated_by_hour(power_data)
```

plot_aggregated_by_month *Plot Aggregated Data by Month*

Description

Generates a bar plot showing monthly input and output sums.

Usage

```
plot_aggregated_by_month(power_data)
```

Arguments

power_data	data frame with timestamp, INPUT, and OUTPUT columns.

Value

A ggplot object showing monthly aggregated values.

Examples

```
# Example using a small sample data frame
power_data <- data.frame(
  timestamp = c(
    as.POSIXct("2000-01-01 01:00:00", tz = "UTC"),
    as.POSIXct("2000-01-02 01:00:00", tz = "UTC"),
    as.POSIXct("2000-02-01 02:00:00", tz = "UTC"),
    as.POSIXct("2000-02-02 02:00:00", tz = "UTC")
  ),
  INPUT = c(1.0, 2.0, 3.0, 4.0),
  OUTPUT = c(4.0, 3.0, 2.0, 1.0)
)
plot_aggregated_by_month(power_data)
```

---

plot_aggregated_by_year

Plot Aggregated Data by Year

Description

Generates a bar plot showing yearly input and output sums.

Usage

```
plot_aggregated_by_year(power_data)
```

Arguments

power_data	data frame with timestamp, INPUT, and OUTPUT columns.

Value

A ggplot object showing yearly aggregated values.

Examples

```
# Example using a small sample data frame
power_data <- data.frame(
  timestamp = c(
    as.POSIXct("2000-01-01 01:00:00", tz = "UTC"),
    as.POSIXct("2000-01-02 01:00:00", tz = "UTC"),
    as.POSIXct("2000-02-01 02:00:00", tz = "UTC"),
    as.POSIXct("2000-02-02 02:00:00", tz = "UTC")
  ),
  INPUT = c(1.0, 2.0, 3.0, 4.0),
  OUTPUT = c(4.0, 3.0, 2.0, 1.0)
)
plot_aggregated_by_year(power_data)
```

---

plot_by_hour_and_month

Plot Hourly Data by Month

Description

Generates a bar plot showing hourly input and output sums by month.

Usage

```
plot_by_hour_and_month(power_data)
```

Arguments

power_data	data frame with timestamp, INPUT, and OUTPUT columns.

Value

A ggplot object showing hourly values by month in facets.

Examples

```
# Example using a small sample data frame
power_data <- data.frame(
  timestamp = c(
    as.POSIXct("2000-01-01 01:00:00", tz = "UTC"),
    as.POSIXct("2000-01-02 01:00:00", tz = "UTC"),
    as.POSIXct("2000-02-01 02:00:00", tz = "UTC"),
    as.POSIXct("2000-02-02 02:00:00", tz = "UTC")
  ),
  INPUT = c(1.0, 2.0, 3.0, 4.0),
  OUTPUT = c(4.0, 3.0, 2.0, 1.0)
)
plot_by_hour_and_month(power_data)
```

plot_heatmap *Generate a Heatmap of Hourly Data by Month*

Description

Creates a heatmap to show hourly input and output data across months.

Usage

```
plot_heatmap(power_data)
```

Arguments

power_data	data frame with timestamp, INPUT, and OUTPUT columns.

Value

A ggplot object with a heatmap representing input/output.

Examples

```
# Example using a small sample data frame
power_data <- data.frame(
  timestamp = c(
    as.POSIXct("2000-01-01 01:00:00", tz = "UTC"),
    as.POSIXct("2000-01-02 01:00:00", tz = "UTC"),
    as.POSIXct("2000-01-01 02:00:00", tz = "UTC"),
    as.POSIXct("2000-01-02 02:00:00", tz = "UTC")
  ),
  INPUT = c(1.0, 2.0, 3.0, 4.0),
  OUTPUT = c(4.0, 3.0, 2.0, 1.0)
)
plot_heatmap(power_data)
```

plot_line_chart *Generate a Line Chart of Hourly Data by Month*

Description

Creates a line chart to show input/output data by hour for each month.

Usage

```
plot_line_chart(power_data)
```

Arguments

power_data	data frame with timestamp, INPUT, and OUTPUT columns.

Value

A ggplot object with a line chart.

Examples

```
# Example using a small sample data frame
power_data <- data.frame(
  timestamp = c(
    as.POSIXct("2000-01-01 01:00:00", tz = "UTC"),
    as.POSIXct("2000-01-02 01:00:00", tz = "UTC"),
    as.POSIXct("2000-01-01 02:00:00", tz = "UTC"),
    as.POSIXct("2000-01-02 02:00:00", tz = "UTC")
  ),
  INPUT = c(1.0, 2.0, 3.0, 4.0),
  OUTPUT = c(4.0, 3.0, 2.0, 1.0)
)
plot_line_chart(power_data)
```

---

plot_map *Map of Dropdown Items to Plot Functions*

Description

A data frame mapping dropdown box items to corresponding plot functions.

Usage

```
plot_map
```

Format

A data frame with dropdown items as row names and plot function names as column values.

---

plot_ridgeline *Generate a Ridgeline Plot of Hourly Data by Month*

Description

Creates a ridgeline plot to show distribution of input/output data by month.

Usage

```
plot_ridgeline(power_data)
```

Arguments

power_data	data frame with timestamp, INPUT, and OUTPUT columns.

Value

A ggplot object with a ridgeline plot.

Examples

```
# Example using a small sample data frame
power_data <- data.frame(
  timestamp = c(
    as.POSIXct("2000-01-01 01:00:00", tz = "UTC"),
    as.POSIXct("2000-01-02 01:00:00", tz = "UTC"),
    as.POSIXct("2000-02-01 01:00:00", tz = "UTC"),
    as.POSIXct("2000-02-02 01:00:00", tz = "UTC")
  ),
  INPUT = c(1.0, 2.0, 3.0, 4.0),
  OUTPUT = c(4.0, 3.0, 2.0, 1.0)
)
plot_ridgeline(power_data)
```

---

plot_stacked_area *Generate a Stacked Area Chart of Hourly Data by Month*

Description

Creates a stacked area chart to visualize input/output data by month.

Usage

```
plot_stacked_area(power_data)
```

Arguments

power_data	data frame with timestamp, INPUT, and OUTPUT columns.

Value

A ggplot object with a stacked area chart.

Examples

```
# Example using a small sample data frame
power_data <- data.frame(
  timestamp = c(
    as.POSIXct("2000-01-01 01:00:00", tz = "UTC"),
    as.POSIXct("2000-01-02 01:00:00", tz = "UTC"),
    as.POSIXct("2000-02-01 02:00:00", tz = "UTC"),
    as.POSIXct("2000-02-02 02:00:00", tz = "UTC")
  ),
  INPUT = c(1.0, 2.0, 3.0, 4.0),
  OUTPUT = c(4.0, 3.0, 2.0, 1.0)
```

```
)
plot_stacked_area(power_data)
```

`plot_top_days` *Generate a Bar Chart of top 10 days of power input vs output*

Description

Creates a Bar Chart and also displays mean and percentile values.

Usage

```
plot_top_days(power_data)
```

Arguments

`power_data`	data frame with `timestamp`, `INPUT`, and `OUTPUT` columns.

Value

A ggplot object with a line chart.

Examples

```
# Example using a small sample data frame
power_data <- data.frame(
  timestamp = c(
    as.POSIXct("2000-01-01 01:00:00", tz = "UTC"),
    as.POSIXct("2000-01-02 01:00:00", tz = "UTC"),
    as.POSIXct("2000-01-01 02:00:00", tz = "UTC"),
    as.POSIXct("2000-01-02 02:00:00", tz = "UTC")
  ),
  INPUT = c(1.0, 2.0, 3.0, 4.0),
  OUTPUT = c(4.0, 3.0, 2.0, 1.0)
)
plot_top_days(power_data)
```

`read_power_data` *Read Power Data*

Description

Reads power data from `.CSV` files in the specified directory, selecting specific columns and using a custom locale for decimal and grouping marks.

Usage

```
read_power_data(data_path)
```

Arguments

`data_path`	Character string specifying the directory containing `.CSV` files to read.

Value

A data frame containing timestamp, input, and output columns from the files.

Examples

```
read_power_data("data/")
```

Index