Package 'timelineR'

December 10, 2019

Title Visualization for Time Series Data

Version 1.0.0

Maintainer Alex Joseph <alex@systeminsights.com>

Description Helps to visualize multi-variate time-series having numeric and factor variables. You can use the package for visual analysis of data by plotting the data for each variable in the desired order and study
interaction between a factor and a numeric variable by creating overlapping plots.

Depends R (>= 3.3.0)

Imports rlang, dplyr, lubridate, stringr, futile.logger, ggplot2, gtable, grid, mtconnectR

Suggests testthat, knitr, rmarkdown

License AGPL-3

Encoding UTF-8

LazyData TRUE

RoxygenNote 6.1.1

VignetteBuilder knitr

NeedsCompilation no

Author Mohsin Vindhani [aut],
Alex Joseph [aut, cre],
Ananthapadmanabhan [aut],
Subramanyam Ravishankar [aut]

Repository CRAN

Date/Publication 2019-12-10 11:30:02 UTC

R topics documented:

`generate_color_mapping` *Generate color mapping for all plots*

Description

When all the plots have the same set of values, instead of typing the color mapping for all plots, default color to value mapping can be given which will generate the color mapping for all the plots based on the unique values present in each column

Usage

```
generate_color_mapping(df, default_color_mapping)
```

Arguments

`df`	Data frame to be plotted using plot_timeline
`default_color_mapping`	Value to color mapping for all the plots in the data frame. ex: default_color_mapping <- c("0" = "#BCBEC0", "1" = "#1279C6")

`match_grep` *Regular expression based extraction*

Description

This function does a regular expression based search for each name in one vector for the values in the other vector and returns a named vector with names as the matched names and values as given in the queried vector.

Usage

```
match_grep(grep_vec, actual_names, use_values = F, return_names = F,
  echo = F)
```

Arguments

`grep_vec`	A named vector with the names to be searched for and the values, which the matching names should hold. It can also be a unnamed vector of names to search for.
`actual_names`	A vector giving the names in which the search is to be made
`use_values`	Logical value. (TRUE) if the values in the grep_vec are to be used for searching. Defualut is FALSE
`return_names`	Logical value (TRUE) if just want to return the matching names and not the values. Defualut is FALSE
`echo`	Logical value(TRUE) To print for each name in the grep_vec, which values in actual_names match and didnt match. Default is FALSE

Value

A named vector with the matched names and substituted values or a vector of macthed names

---

`plot_timeline` *Plotting function (standard style)*

Description

Plots time series data of State type (factors) as stripe charts, Numeric data type as step charts and an overlapping combination of a plot of State type and Numeric type .

Usage

```
plot_timeline(timeline_df, data_cols = NULL, start_time = NULL,
  end_time = NULL, ylimits = NULL, scale_vals = NULL,
  titles = NULL, ylabels = NULL, save_path = NULL,
  add_legend = TRUE, plot_size_ratios = NULL,
  overlap_plots_names = NULL, color_mapping = list(),
  order_plots = NULL, plot_output = T, numeric_plot_type = "line",
  output_width = 1500, output_height = 800)
```

Arguments

`timeline_df`	Dataframe
`data_cols`	A vector showing the columns to subset for plotting
`start_time`	is left end point of the plot e.g: start_time="2014-01-30 09:53:02.792663 UTC" or start_time=1391075599
`end_time`	is right end point of the plot
`ylimits`	A named vector to determine the limits on the y-axis for Sample plots e.g: ylimits=list(a=c(0,100),d=c(-100,50)). The names must be present in the data frame
`scale_vals`	A named vector to scale numeric data e.g: scale_vals = c(a=10), matching data will be multiplied by 10
`titles`	A named vector to give titles to the plot. For state and numeric plots, the names should be the same as in the data frame. For overlapping plots, it should be the same as the name given in the overlap_plots_names. e.g: titles = c(ab="first plot",cd="second plot")
`ylabels`	change the labels on y-axis of plots e.g: ylabel=c(ab="value",bcd="tmeperature")
`save_path`	if a file_path is specified, then the image will be saved to that location.
`add_legend`	TRUE (default) if legend is needed for the plots
`plot_size_ratios`	proportion of event plot size to the sample plot size

`overlap_plots_names`
: specify the data items to be overlapped. Plots of the same type can only be overlapped for now. This argument can be used to specify the order of plots. e.g.: overlap_plots="list(overlap_plot1 = c(state1,numeric1), overlap_plot2 = c(state1,numeric2)"

`color_mapping`
: A named list of named vectors. The names of the list are the names of the state columns in the data frame. Each named vector for a state should have color mapping for all the states in the column.

`order_plots`
: A vector containing the name of the plots to be plotted. The plots in the final output are arranged according to the order of the names in this vector.

`plot_output`
: Logical argument to specify if the output is required to be plotted or not. TRUE(default)

`numeric_plot_type`
: The plot type for numeric variables. It can be either of the type 'line','step' or 'point'. By default the type is 'line'.

`output_width`
: The width of the plot while saving. The value is in pixels.

`output_height`
: The height of the plot while saving. The value is in pixels.

Value

A grob of all the plots

Index