Package ‘chartql’

April 4, 2019

Type Package
Title Simplified Language for Plots and Charts
Version 0.1.0
Imports ggplot2 (>= 2.1.0), stringr, stats
Maintainer Rohail Syed <rohailsyed@gmail.com>
Description Provides a very simple syntax for the user to generate custom plot(s) without having to remember complicated 'ggplot2' syntax. The 'chartql' package uses 'ggplot2' and manages all the syntax complexities internally. As an example, to generate a bar chart of company sales faceted by product category further faceted by season of the year, we simply write: ``CHART bar X category, season Y sales``.
License GPL-3
Encoding UTF-8
URL https://github.com/rmsyed/chartql
LazyData true
NeedsCompilation no
Author Rohail Syed [aut, cre]
Repository CRAN
Date/Publication 2019-04-04 15:40:03 UTC

R topics documented:
cql ................................................................. 2

Index 5
cql

chartql Chart Generator

Description

Generates a chart/plot using source_frame as the source dataframe and using cql_string as the query string. Options and structure of the query string are described below:

Usage

cql(source_frame, cql_string)

Arguments

source_frame DataFrame object. Used as the source to build the plot.
cql_string chartql query string. Used to specify options in terms of how to build the plot. This includes the type of plot, choice of x and y variables, axis titles and more. Full parameter list below.

Details

The chartql query language uses a fairly simple format to let users generate amazing plots but with minimal scripting, allowing for much faster prototyping. The chartql string uses the following general format:

\[
\text{CHART <chart_type> X <x_var> Y <y_var> (Options) <options>}
\]

The main variables that are required for all types of plots are:

- **Chart <chart_type>:** bar|scatter|hist|line
- **X <x_var>, <(x_facet)>:** X-axis variable. May include a second categorical variable (comma separated)
- **Y <y_var>:** Y-axis variable. *Not used for hist.

Optional Variables

- **X_label <xlab>:** X-axis custom label. Defaults to <x_var>
- **Y_label <ylab>:** Y-axis custom label. Defaults to <y_var>
- **Legend <legend>:** Legend custom label. Defaults to <x_category>
- **Colorset <clist>:** Custom set of colors for <x_category> levels. Must be comma-separated list.
- **AggFunc <func>:** Summary function for bar type. Valid types: mean|median|count|sum. Defaults to mean.
- **Fit <show_se>:** Valid types: true|false. Fit quadratic line. Show/Don’t show standard error curves.
- **ConfInt <interval>:** Show error bars. Value is confidence interval value. E.g. ’.95’
**Value**

- **response**  
  List object. Entries are the valid parameters and their values extracted from `cql_string`.

- **plot_obj**  
  ggplot object. This is the resulting plot.

**Note**

1. Custom label parameters `x_label`, `y_label` and `legend` may contain newlines but should be escaped with an extra backslash as: "\n".

2. The parameter names themselves are not case-sensitive (e.g. both "CHART bar" and "chart bar" are valid formatting).

3. You can hide the Legend by setting the value to "<" (e.g. "Legend ' '<")

**Examples**

```r
# Test data
dframe <- data.frame(
category = factor(c(rep("Sports",50), rep("Home", 150), rep("Fashion", 100)),
season = factor(c(rep("Fall","Winter"),150)),
sales = c(runif(100,min=0,max=100), runif(100,min=50,max=200), runif(100,min=50,max=80))
)
dframe$visitors <- dframe$sales * runif(300, min=0.5, max=1.5)

# Bar chart with product category on x-axis and total sales on y-axis
cql_str <- "CHART bar X category Y sales";
cql(dframe, cql_str);

# Bar chart but facet x-axis with both category and season of the year
cql_str <- "CHART bar X category, season Y sales X_label 'Product\nCategory'";
cql(dframe, cql_str);

# Bar chart with 95% confidence interval error bars
`cql_str <- "CHART bar X category, season Y sales ConfInt '.95'";
cql(dframe, cql_str);

# Bar chart but specify the colors for each season
`cql_str <- "CHART bar X category, season Y sales Colorset '#FF900, #90099'"
cql(dframe, cql_str);

# Scatter plot of number of visitors and sales
`cql_str <- "CHART scatter X visitors Y sales";
cql(dframe, cql_str);
```
# Scatter plot but facet by season

cql_str <- "CHART scatter X visitors, season Y sales";
cql(dframe, cql_str);

# Scatter plot with fitted line (no SE curves)
cql_str <- "CHART scatter X visitors, season Y sales Fit 'false'";
cql(dframe, cql_str);
Index

cql, 2