Package ‘ggplotlyExtra’

October 13, 2022

Title Extra Convenience Functions for 'Plotly'

Version 0.0.1

Author Omar I. Elashkar <omar.ibrahim@miuegypt.edu.eg>

Maintainer Omar I. Elashkar <omar.ibrahim@miuegypt.edu.eg>

Depends R (>= 3.5.0)

Description Convenience functions for smooth conversion from 'ggplot' to 'plotly' where the conversion using ggplotly() usually gives an unexpected labels. The package ease the process of making a 'plotly' figures generated from 'ggplot2' object more aesthetic in terms of labels and customizability.

Imports ggplot2, plotly, rlang

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.0.0

NeedsCompilation no

Repository CRAN

Date/Publication 2019-12-02 16:20:06 UTC

R topics documented:

ggplotly_histogram *Clean 'ggplot2' Histogram to be Converted to 'Plotly'*

Description

Create 'ggplot2' histogram that translate nicely to 'plotly'.

Usage

```
ggplotly_histogram(data = NULL, mapping = NULL, position = "stack",
  ..., binwidth = NULL, na.rm = FALSE, show.legend = NA)
```

Arguments

`data` The data to be displayed in this layer. There are three options:

If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot()`.

A `data.frame`, or other object, will override the plot data. All objects will be fortified to produce a data frame. See `fortify()` for which variables will be created.

A `function` will be called with a single argument, the plot data. The return value must be a `data.frame`, and will be used as the layer data. A `function` can be created from a `formula` (e.g. `~ head(.x, 10)`).

`mapping` Set of aesthetic mappings created by `aes()` or `aes_()`. If specified and `inherit.aes = TRUE` (the default), it is combined with the default mapping at the top level of the plot. You must supply `mapping` if there is no plot mapping.

`position` Position adjustment, either as a string, or the result of a call to a position adjustment function.

`...` Other arguments passed on to `layer()`. These are often aesthetics, used to set an aesthetic to a fixed value, like `colour = "red"` or `size = 3`. They may also be parameters to the paired geom/stat.

`binwidth` The width of the bins. Can be specified as a numeric value or as a function that calculates width from unscaled x. Here, "unscaled x" refers to the original x values in the data, before application of any scale transformation. When specifying a function along with a grouping structure, the function will be called once per group. The default is to use `bins` bins that cover the range of the data. You should always override this value, exploring multiple widths to find the best to illustrate the stories in your data.

The bin width of a date variable is the number of days in each time; the bin width of a time variable is the number of seconds.

`na.rm` If `FALSE`, the default, missing values are removed with a warning. If `TRUE`, missing values are silently removed.

`show.legend` logical. Should this layer be included in the legends? `NA`, the default, includes if any aesthetics are mapped. `FALSE` never includes, and `TRUE` always includes. It can also be a named logical vector to finely select the aesthetics to display.

Details

‘ggplotly_histogram()‘ is a function that is used to create a ’ggplot2’ histogram, yet on conversion to ’plotly’ using ’ggplotly()’, the resulted plot will hold the correct labeling information, which are "Range", "Count" and "Density".

Value

ggplot bar layer

Examples

```
library(ggplot2)
library(plotly)
# create the histogram using `ggplotly_histogram()`
p <- ggplot() + ggplotly_histogram(data = ToothGrowth, mapping = aes(len))+
xlab("len")

# convert `ggplot` object to `plotly` object
ggplotly(p, tooltip = c("Range", "count", "density"))
```

Index