

Package ‘leaflethex’

September 4, 2019

Title Hexbin Layers for 'Leaflet' Maps

Version 0.1.0

Description Adds functionality for hexbin layers to 'leaflet' maps. There is a javascript library for this hexbin function already so this package creates an R interface for using that library to make it much easier to access. This package is intended to be a plugin for 'leaflet' and the API attempts to mirror 'leaflet' style.

License GPL-3

Encoding UTF-8

LazyData true

Depends leaflet, htmltools, htmlwidgets, stats

Imports dplyr, readr, tibble

Suggests spelling, knitr, rmarkdown, tinytex

Language en-US

RoxygenNote 6.1.1

VignetteBuilder knitr

NeedsCompilation no

Author Randall Pruim [aut] (<<https://orcid.org/YOUR-ORCID-ID>>),
Jason Klaassen [aut, cre]

Maintainer Jason Klaassen <jasonucc12@gmail.com>

Repository CRAN

Date/Publication 2019-09-04 08:20:02 UTC

R topics documented:

| | |
|-------------------------|---|
| addHexbin | 2 |
| pluginFactory | 3 |

| | |
|--------------|----------|
| Index | 5 |
|--------------|----------|

 addHexbin

Hexbin Layers for Leaflet Plots

Description

Create hexbin layers for leaflet plots.

Usage

```
addHexbin(map, data = NULL, radius = 12, opacity = 0.5,
  duration = 500, lowEndColor = "white", highEndColor = "blue",
  uniformSize = FALSE, uniformColor = NULL,
  sizeSummaryFunction = c("count", "sum", "max", "min", "mean",
    "median"), sizevar = NULL, colorSummaryFunction = c("count", "sum",
    "max", "min", "mean", "median"), colorvar = NULL)
```

Arguments

| | |
|----------------------|--|
| map | The leaflet map object to apply the hexbin layer to. Makes this function compatible with the %>% operator |
| data | data frame or tibble - alternate data to use for this hexbin instead of default map data |
| radius | choose the base size for the hexagons |
| opacity | decimal between 0.0 and 1.0 - choose the percent of opacity for the hexagons |
| duration | positive integer milliseconds that the animation takes for drawing the hexagons |
| lowEndColor | choose the color for the smaller hexagons |
| highEndColor | choose the color for the larger hexagons |
| uniformSize | a logical indicating whether all hexagons should be the same size. |
| uniformColor | a color that overrides lowEndColor and highEndColor to make the color uniform across the hexagon sizes. |
| sizeSummaryFunction | a string that specifies which summary function to use on sizevar to modulate the size of the hexagons. The options are 'count', 'sum', 'max', 'min', 'mean', and 'median'. |
| sizevar | a string that specifies which variable in the user specified data frame will be used to calculate the size of the hexagons. |
| colorSummaryFunction | a string that specifies which summary function to use on colorvar to modulate the color of the hexagons. The options are 'count', 'sum', 'max', 'min', 'mean', and 'median'. |
| colorvar | a string that specifies which variable in the user specified data frame will be used to calculate the color of the hexagons. |

Value

map parameter, but with the hexbinLayer attached so that it can be used with the %>% pipe operator

Note

Do not use uniformColor and uniformSize together as it will not give any insights to the data

If colorSummaryFunction and colorvar are not specified, the color will mirror the sizevar unless uniform color set to TRUE.

See Also

[A JSFiddler Hexbin example](#) by Ryan

Examples

```
leaflet::leaflet(data.frame(lat = 42.9634 + rnorm(1000), lng = -85.6681 + rnorm(1000))) %>%
  addTiles() %>% addHexbin()
```

```
leaflet::leaflet(data.frame(lat = 42.9634 + rnorm(1000), lng = -85.6681 + rnorm(1000))) %>%
  addTiles() %>% addHexbin(radius=25, lowEndColor='purple', highEndColor='orange')
```

 pluginFactory

Create plugin functions to implement Leaflet pure-JS libraries in R

Description

pluginFactory() takes in a couple js files (code and dependencies) and creates a plugin function that applies this plugin to a given map. The new function also returns the map so it can be piped with the leaflet functions

Usage

```
pluginFactory(name = "JSPlugin", location, jsfilename, dependencies,
  stylesheet = NULL)
```

Arguments

| | |
|--------------|--|
| name | A name for your plugin |
| location | The parent folder of your js file, dependency file, and css stylesheet |
| jsfilename | A .js filename relative to the parent folder at location that holds the source code for modifying the map object |
| dependencies | A .js filename(s) of all other js libraries the source code depends on |
| stylesheet | A .css stylesheet for adding styles to the entire page |

Value

A function with arguments `map`, `data`, and ... to be used as a plugin to leaflet just like the `leaflet::addCircles()` function

See Also

Some code is used from [this github gist by Joe Chang](#)

Examples

```
df <- data.frame(
  lat = 42.9634 + stats::rnorm(100),
  lng = -85.6681 + stats::rnorm(100),
  size = runif(100, 5, 20),
  color = sample(colors(), 100)
)
addJS <-
  pluginFactory(
    "Some JS Plugin",
    system.file("js", "", package = "leaflethex"), "hexbin.js", "deps.js", stylesheet="hexbin.css")

leaflet(df, width = 600, height = 300) %>%
  addTiles() %>%
  addJS(radius = 20, highEndColor = "yellow")
```

Index

`addHexbin`, 2

`pluginFactory`, 3