

# Package ‘leafletZH’

October 25, 2024

**Type** Package

**Title** Adds a Chinese Choropleth Leaflet Map

**Version** 0.1.0

**Maintainer** Damonsoul <chenmaowei96@gmail.com>

**Description** Provides 'sf' data for Chinese provinces and cities, methods for plotting shape maps of Chinese provinces and cities, and a layer for 'leaflet' with Gaode tiles.  
It is designed to facilitate geographical data visualization in China.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Imports** geojsonsf, htmltools, htmlwidgets, leaflet, leaflet.extras, sf, stringr

**RoxygenNote** 7.3.2

**Depends** R (>= 4.0.0)

**NeedsCompilation** no

**Author** Damonsoul [aut, cre]

**Repository** CRAN

**Date/Publication** 2024-10-25 08:40:02 UTC

## Contents

addcityShape . . . . .	2
addProvinceShape . . . . .	4
addTilesAmap . . . . .	7
china_city . . . . .	8
china_province . . . . .	8

<b>Index</b>	<b>9</b>
--------------	----------

---

addcityShape	<i>Adds a choropleth map layer for cities with additional customization options.</i>
--------------	--

---

### Description

Adds a choropleth map layer for cities with additional customization options.

### Usage

```
addcityShape(  
  map,  
  data,  
  adcode = NULL,  
  layerId = NULL,  
  group = NULL,  
  valueProperty = NULL,  
  labelProperty = NULL,  
  labelOptions = leaflet::labelOptions(),  
  popupProps = NULL,  
  popupOptions = leaflet::popupOptions(),  
  scale = c("white", "red"),  
  steps = 5,  
  mode = "q",  
  channelMode = c("rgb", "lab", "hsl", "lch"),  
  padding = NULL,  
  correctLightness = FALSE,  
  bezierInterpolate = FALSE,  
  colors = NULL,  
  stroke = TRUE,  
  color = "#ffffff",  
  weight = 1,  
  opacity = 0.5,  
  fillOpacity = 0.7,  
  dashArray = NULL,  
  smoothFactor = 1,  
  noClip = FALSE,  
  pathOptions = leaflet::pathOptions(),  
  highlightOptions = leaflet::highlightOptions(weight = 2, color = "#000000", fillOpacity  
    = 1, opacity = 1, bringToFront = TRUE, sendToBack = TRUE),  
  legendOptions = NULL,  
  ...  
)
```

### Arguments

map	The leaflet map object to add the layer to.
-----	---

data	A data frame containing the data to be visualized.
adcode	China administrative division code
layerId	An optional string to identify the layer.
group	An optional string for grouping data.
valueProperty	The property in the geojson data that corresponds to the value to be mapped.
labelProperty	The property in the geojson data that will be used for labels.
labelOptions	Options for labels, defaults to leaflet's labelOptions.
popupProps	A named vector of properties to display in the popup.
popupOptions	Options for popups, defaults to leaflet's popupOptions.
scale	A vector of colors to use for the scale of the choropleth map.
steps	The number of steps for the color scale.
mode	The mode for the color scale, can be "q" for quantile, "e" for equal interval, etc.
channelMode	The color channel mode, can be "rgb", "lab", "hsl", or "lch".
padding	Optional padding for the choropleth layer.
correctLightness	A logical value to correct lightness for color scales.
bezierInterpolate	Whether to use bezier interpolation for the lines.
colors	An optional vector of colors to override the default color scale.
stroke	Whether to draw the stroke along the paths.
color	The color for the paths, defaults to white.
weight	The weight for the paths.
opacity	The opacity for the paths.
fillOpacity	The fill opacity for the paths.
dashArray	An optional array to create dashed lines.
smoothFactor	A factor to smooth the factor for the paths.
noClip	Whether to disable clipping of the paths.
pathOptions	Additional options for the paths, defaults to leaflet's pathOptions.
highlightOptions	Options for highlighting features, defaults to leaflet's highlightOptions.
legendOptions	Options for the legend.
...	Additional arguments passed to other functions.

**Value**

The modified leaflet map object with the added layer.

**Examples**

```
# use adcode,adcode can be obtained from leafletZH::china_city
library(leaflet)
library(leaflet.extras)
library(leafletZH)
library(sf)
data <- data.frame(adcode = seq(110101, 110110, 1), value = runif(5))
leaflet() |>
  leafletZH::addTilesAmap() |>
  addcityShape(
    data = data, adcode = "adcode", valueProperty = "value",
    popupProps = c("value")
  ) |>
  setView(lng = 116, lat = 40, zoom = 8)
```

---

addProvinceShape	<i>Adds a choropleth map layer for provinces with additional customization options.</i>
------------------	---

---

**Description**

Adds a choropleth map layer for provinces with additional customization options.

**Usage**

```
addProvinceShape(
  map,
  data,
  adcode = NULL,
  provinceName = NULL,
  layerId = NULL,
  group = NULL,
  valueProperty = NULL,
  labelProperty = NULL,
  labelOptions = leaflet::labelOptions(),
  popupProps = NULL,
  popupOptions = leaflet::popupOptions(),
  scale = c("white", "red"),
  steps = 5,
  mode = "q",
  channelMode = c("rgb", "lab", "hsl", "lch"),
  padding = NULL,
  correctLightness = FALSE,
  bezierInterpolate = FALSE,
  colors = NULL,
```

```

    stroke = TRUE,
    color = "#ffffff",
    weight = 1,
    opacity = 0.5,
    fillOpacity = 0.7,
    dashArray = NULL,
    smoothFactor = 1,
    noClip = FALSE,
    pathOptions = leaflet::pathOptions(),
    highlightOptions = leaflet::highlightOptions(weight = 2, color = "#000000", fillOpacity
      = 1, opacity = 1, bringToFront = TRUE, sendToBack = TRUE),
    legendOptions = NULL,
    ...
  )

```

### Arguments

map	The leaflet map object to add the layer to.
data	A data frame containing the data to be visualized.
adcode	China administrative division code
provinceName	A string specifying the column name in the data frame that corresponds to the province names.
layerId	An optional string to identify the layer.
group	An optional string for grouping data.
valueProperty	The property in the geojson data that corresponds to the value to be mapped.
labelProperty	The property in the geojson data that will be used for labels.
labelOptions	Options for labels, defaults to leaflet's labelOptions.
popupProps	A named vector of properties to display in the popup.
popupOptions	Options for popups, defaults to leaflet's popupOptions.
scale	A vector of colors to use for the scale of the choropleth map.
steps	The number of steps for the color scale.
mode	The mode for the color scale, can be "q" for quantile, "e" for equal interval, etc.
channelMode	The color channel mode, can be "rgb", "lab", "hsl", or "lch".
padding	Optional padding for the choropleth layer.
correctLightness	A logical value to correct lightness for color scales.
bezierInterpolate	Whether to use bezier interpolation for the lines.
colors	An optional vector of colors to override the default color scale.
stroke	Whether to draw the stroke along the paths.
color	The color for the paths, defaults to white.
weight	The weight for the paths.

opacity	The opacity for the paths.
fillOpacity	The fill opacity for the paths.
dashArray	An optional array to create dashed lines.
smoothFactor	A factor to smooth the factor for the paths.
noClip	Whether to disable clipping of the paths.
pathOptions	Additional options for the paths, defaults to leaflet's pathOptions.
highlightOptions	Options for highlighting features, defaults to leaflet's highlightOptions.
legendOptions	Options for the legend.
...	Additional arguments passed to other functions.

**Value**

The modified leaflet map object with the added layer.

**Examples**

```
# Plot using province name, match using first two words of field
library(leaflet)
library(leaflet.extras)
library(leafletZH)
data <- data.frame(name = leafletZH::china_province$name, value = runif(34))
backg <- htmltools::tags$style(".leaflet-container { background: #000; }")
leaflet() |>
  addProvinceShape(
    data = data, provinceName = "name", valueProperty = "value",
    popupProps = c("value")
  ) |>
  setView(lng = 110, lat = 40, zoom = 4) |>
  htmlwidgets::prependContent(backg)
```

```
# Use adcode to match, adcode can be obtained in leafletZH::china_province
library(leaflet)
library(leaflet.extras)
library(leafletZH)
library(sf)
data <- data.frame(adcode = seq(110000, 150000, 10000), value = runif(5))
leaflet() |>
  leafletZH::addTilesAmap() |>
  addProvinceShape(
    data = data, adcode = "adcode", valueProperty = "value",
    popupProps = c("value")
  ) |>
  setView(lng = 110, lat = 40, zoom = 4)
```

---

addTilesAmap	<i>Adds a tile layer from Amap to a leaflet map.</i>
--------------	--

---

### Description

This function adds a tile layer from Amap to a leaflet map object.

### Usage

```
addTilesAmap(  
  map,  
  attribution = "&copy; <a href=\"http://amap.com\">amap.com</a >",  
  ...  
)
```

### Arguments

map	A leaflet map object to which the tile layer will be added.
attribution	A string containing the attribution text to be displayed on the map. It defaults to "&copy; <a href=\"http://amap.com\">amap.com</a>".
...	Additional arguments to be passed to the 'leaflet::addTiles' function.

### Value

The leaflet map object with the added tile layer.

### Examples

```
library(leaflet)  
leaflet() %>%  
  addTilesAmap() %>%  
  setView(  
    lng = 120.33739,  
    lat = 31.13533,  
    zoom = 3  
  )
```

---

china_city	<i>city</i>
------------	-------------

---

**Description**

city data for china source: <http://datav.aliyun.com/tools/atlas/>

**name** city name

**adcode** adcode ...

**Usage**

china\_city

**Format**

An object of class sf (inherits from data.frame) with 476 rows and 10 columns.

---

china_province	<i>province</i>
----------------	-----------------

---

**Description**

province data for china source: <http://datav.aliyun.com/tools/atlas/>

**name** province name

**adcode** adcode ...

**Usage**

china\_province

**Format**

An object of class sf (inherits from data.frame) with 34 rows and 10 columns.



# Index

## \* datasets

china\_city, 8

china\_province, 8

addcityShape, 2

addProvinceShape, 4

addTilesAmap, 7

china\_city, 8

china\_province, 8