Package 'choroplethr'

June 21, 2018

Title Simplify the Creation of Choropleth Maps in R

Description Choropleths are thematic maps where geographic regions, such as states, are colored according to some metric, such as the number of people who live in that state. This package simplifies this process by 1. Providing ready-made functions for creating choropleths of common maps. 2. Providing data and API connections to interesting data sources for making choropleths. 3. Providing a framework for creating choropleths from arbitrary shapefiles. 4. Overlaying those maps over reference maps from Google Maps.

Version 3.6.2

Author Ari Lamstein <ari@lamsteinconsulting.com>[cre, aut], Brian P Johnson <brian@pjohnson.info> [ctb, frontend animation code]

Maintainer Ari Lamstein <ari@lamsteinconsulting.com>

URL www.choroplethr.com

License BSD_3_clause + file LICENSE

Imports scales, Hmisc, stringr, ggplot2 (>= 2.0.0), dplyr, R6, WDI, ggmap, RgoogleMaps, tigris, gridExtra

Suggests testthat, choroplethrMaps, choroplethrAdmin1 (>= 1.1.0)

Depends R (>= 3.0.0), acs

Collate 'acs.R' 'choropleth.R' 'admin1.R' 'admin1_region.R' 'choroplethr_animate.R' 'choroplethr_wdi.R' 'country.R' 'usa.R' 'county.R' 'county_zoom.R' 'data.R' 'deprecated.R' 'get_county_demographics.R' 'get_state_demograhpics.R' 'get_tract_demographics.R' 'startup_messages.R' 'state.R' 'tracts.R' 'utils.R'

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-06-21 07:42:29 UTC

R topics documented:

Admin1Choropleth *An R6 object for creating Administration Level 1 choropleths.*

Description

An R6 object for creating Administration Level 1 choropleths.

Usage

```
Admin1Choropleth
```

Format

An object of class `R6ClassGenerator` of length 24.

Admin1RegionChoropleth *An R6 object for creating Administration Level 1 choropleths based on regions.*

Description

Compare with the Admin1Choropleth object, which creates Admin 1 choropleths based on Countries. This function is useful if you want a map that spans multiple countries - Especially if it only needs to include a portion of a country.

Usage

```
Admin1RegionChoropleth
```

Format

An object of class `R6ClassGenerator` of length 24.

admin1_choropleth *Create an admin1-level choropleth for a specified country*

Description

The map used comes from ?admin1.map in the choroplethrAdmin1 package. See ?get_admin_countries and ?get_admin_regions in the choroplethrAdmin1 package for help with the spelling of regions.

Usage

```
admin1_choropleth(country.name, df, title = "", legend = "",
  num_colors = 7, zoom = NULL, reference_map = FALSE)
```

Arguments

country.name	The name of the country. Must exactly match how the country is named in the "country" column of ?admin1.regions in the choroplethrAdmin1 package.
df	A data.frame with a column named "region" and a column named "value". Elements in the "region" column must exactly match how regions are named in the "region" column in ?admin1.regions in the choroplethrAdmin1 package
title	An optional title for the map.
legend	An optional name for the legend.
num_colors	The number of colors on the map. A value of 1 will use a continuous scale. A value in [2, 9] will use that many colors.
zoom	An optional vector of regions to zoom in on. Elements of this vector must exactly match the names of regions as they appear in the "region" column of ?admin1.regions.
reference_map	If true, render the choropleth over a reference map from Google Maps.

Examples

```
## Not run:

library(choroplethrAdmin1)

data(df_japan_census)
head(df_japan_census)
# set the value we want to map to be the 2010 population estimates
df_japan_census$value=df_japan_census$pop_2010

# default map of all of japan
admin1_choropleth("japan",
                  df_japan_census,
                  "2010 Japan Population Estimates",
                  "Population")

# zoom in on the Kansai region and use a continuous scale
```

```
kansai = c("mie", "nara", "wakayama", "kyoto", "osaka", "hyogo", "shiga")
admin1_choropleth("japan",
                  df_japan_census,
                  "2010 Japan Population Estimates",
                  "Population",
                  1,
                  kansai)

admin1_choropleth("japan",
                  df_japan_census,
                  "2010 Japan Population Estimates",
                  "Population",
                  1,
                  kansai,
                  reference_map = TRUE)

## End(Not run)
```

---

admin1_region_choropleth

Create a map of Administrative Level 1 regions

Description

Unlike ?admin1_choropleth, the regions here can span multiple countries.

Usage

```
admin1_region_choropleth(df, title = "", legend = "", num_colors = 7,
  zoom = NULL, reference_map = FALSE)
```

Arguments

df	A data.frame with a column named "region" and a column named "value". Elements in the "region" column must exactly match how regions are named in the "region" column in ?admin1.regions in the choroplethrAdmin1 package
title	An optional title for the map.
legend	An optional name for the legend.
num_colors	The number of colors on the map. A value of 1 will use a continuous scale. A value in [2, 9] will use that many colors.
zoom	An optional vector of regions to zoom in on. Elements of this vector must exactly match the names of regions as they appear in the "region" column of ?admin1.regions.
reference_map	If true, render the choropleth over a reference map from Google Maps.

Details

The map used comes from ?admin1.map in the choroplethrAdmin1 package. See ?get_admin_countries and ?get_admin_regions in the choroplethrAdmin1 package for help with the spelling of regions.

Examples

```
## Not run:

library(choroplethrAdmin1)

# map of continental us + southern canada

data("continental_us_states")
lower_canada = c("british columbia", "alberta", "saskatchewan", "manitoba", "ontario", "quebec")
regions = c(lower_canada, continental_us_states)
df = data.frame(region=regions, value=sample(1:length(regions)))

admin1_region_choropleth(df)


## End(Not run)
```

---

calculate_percent_change

Calculate the percentage change between two choroplethr dataframes.

Description

Merges df1 and df2 on column named "region", and computes percentage change from df1$value to df2$value. Result is in the new "value" column, and rounded to two digits.

Usage

```
calculate_percent_change(df1, df2)
```

Arguments

df1	A dataframe with columns named "region" and "value"
df2	A dataframe with columns named "region" and "value"

Examples

```
# load median age estimates from 2010 and 2015
data(df_state_age_2010)
data(df_state_age_2015)

df_age_diff = calculate_percent_change(df_state_age_2010, df_state_age_2015)
state_choropleth(df_age_diff,
```

```
title      = "Percent Change in Median Age, 2010-2015",
legend     = "Percent Change",
num_colors = 0)
```

`Choropleth` *The base Choropleth object.*

Description

The base Choropleth object.

Usage

```
Choropleth
```

Format

An object of class `R6ClassGenerator` of length 24.

`choroplethr` *Create a choropleth*

Description

This function is deprecated as of choroplethr version 2.0.0. Please use ?state_choropleth, ?county_choropleth, ?zip_map and ?country_choroplethr instead. The last version of choroplethr in which this function worked was version 1.7.0, which can be downloaded from CRAN here: http://cran.r-project.org/web/packages/choroplethr/in

Usage

```
choroplethr(...)
```

Arguments

...	All arguments are ignored.

choroplethr_acs *Create a choropleth from ACS data.*

Description

This function is deprecated as of choroplethr version 3.0.0. Please use ?state_choropleth_acs, ?county_choropleth_acs, ?zip_choropleth_acs. The last version of choroplethr in which this function worked was version 2.1.1, which can be downloaded from CRAN here: http://cran.r-project.org/web/packages/choropleth

Usage

```
choroplethr_acs(...)
```

Arguments

...	All arguments are ignored.

choroplethr_animate *Animate a list of choropleths*

Description

Given a list of choropleths, represented as ggplot2 objects

1. Save the individual images to the working directory with the naming convention "choropleth_1.png", "choropleth_2.png", etc.
2. Write a file called "animated_choropleth.html" which contains a viewer which animates them.

Usage

```
choroplethr_animate(choropleths)
```

Arguments

choropleths	A list of choropleths represented as ggplot2 objects.

Value

Nothing. However, a variable number of files are written to the current working directory.

Author(s)

Ari Lamstein (R code) and Brian Johnson (JavaScript, HTML and CSS code)

Examples

```
## Not run:
data(df_president_ts)
?df_president_ts # time series of all US presidential elections 1789-2012

# create a list of choropleths of presidential election results for each year
choropleths = list()
for (i in 2:(ncol(df_president_ts))) {
  df           = df_president_ts[, c(1, i)]
  colnames(df) = c("region", "value")
  title        = paste0("Presidential Election Results: ", colnames(df_president_ts)[i])
  choropleths[[i-1]] = state_choropleth(df, title=title)
}

# set working directory and animate
setwd("~/Desktop")
choroplethr_animate(choropleths)

## End(Not run)
```

`choroplethr_wdi` *Create a country-level choropleth using data from the World Bank's World Development Indicators (WDI)*

Description

Create a country-level choropleth using data from the World Bank's World Development Indicators (WDI)

Usage

```
choroplethr_wdi(code = "SP.POP.TOTL", year = 2012, title = "",
  num_colors = 7, zoom = NULL)
```

Arguments

`code`	The WDI code to use.
`year`	The year of data to use.
`title`	A title for the map. If not specified, automatically generated to include WDI code and year.
`num_colors`	The number of colors to use on the map. A value of 1 will use a continuous scale, and a value in [2, 9] will use that many colors.
`zoom`	An optional list of countries to zoom in on. Must come from the "name" column in ?country.regions.

Value

A choropleth.

References

Uses the WDI function from the WDI package by Vincent Arel-Bundock.

Examples

```
## Not run:
# See http://data.worldbank.org/indicator/SP.POP.TOTL
choroplethr_wdi(code="SP.POP.TOTL", year=2012, title="2012 Population Estimates", num_colors=1)

# See http://data.worldbank.org/indicator/SP.DYN.LE00.IN
choroplethr_wdi(code="SP.DYN.LE00.IN", year=2012, title="2012 Life Expectancy Estimates")

# See http://data.worldbank.org/indicator/NY.GDP.PCAP.CD
choroplethr_wdi(code="NY.GDP.PCAP.CD", year=2012, title="2012 Per Capita Income")

## End(Not run)
```

---

continental_us_states *A vector of the names of US Continental US States.*

Description

A vector of the names of US Continental US States.

Usage

```
data(continental_us_states)
```

Author(s)

Ari Lamstein

CountryChoropleth *An R6 object for creating country-level choropleths.*

Description

An R6 object for creating country-level choropleths.

Usage

```
CountryChoropleth
```

Format

An object of class `R6ClassGenerator` of length 24.

`country_choropleth` *Create a country-level choropleth*

Description

The map used is country.map in the choroplethrMaps package. See country.regions for an object which can help you coerce your regions into the required format.

Usage

```
country_choropleth(df, title = "", legend = "", num_colors = 7,
  zoom = NULL)
```

Arguments

`df`	A data.frame with a column named "region" and a column named "value". Elements in the "region" column must exactly match how regions are named in the "region" column in ?country.map.
`title`	An optional title for the map.
`legend`	An optional name for the legend.
`num_colors`	The number of colors to use on the map. A value of 0 uses a divergent scale (useful for visualizing negative and positive numbers), A value of 1 uses a continuous scale (useful for visualizing outliers), and a value in [2, 9] will use that many quantiles.
`zoom`	An optional vector of countries to zoom in on. Elements of this vector must exactly match the names of countries as they appear in the "region" column of ?country.regions

Examples

```
# demonstrate default options
data(df_pop_country)
country_choropleth(df_pop_country, "2012 World Bank Populate Estimates")

# demonstrate continuous scale
country_choropleth(df_pop_country, "2012 World Bank Populate Estimates", num_colors=1)

# demonstrate zooming
country_choropleth(df_pop_country,
                   "2012 World Bank Population Estimates",
                   num_colors=1,
                   zoom=c("united states of america", "canada", "mexico"))
```

CountyChoropleth *Create a county-level choropleth*

Description

Create a county-level choropleth

Usage

```
CountyChoropleth
```

Format

An object of class `R6ClassGenerator` of length 24.

CountyZoomChoropleth *Create a county-level choropleth that zooms on counties, not states.*

Description

Create a county-level choropleth that zooms on counties, not states.

Usage

```
CountyZoomChoropleth
```

Format

An object of class `R6ClassGenerator` of length 24.

county_choropleth *Create a choropleth of US Counties*

Description

The map used is county.map in the choroplethrMaps package. See country.regions in the choroplethrMaps package for an object which can help you coerce your regions into the required format.

Usage

```
county_choropleth(df, title = "", legend = "", num_colors = 7,
  state_zoom = NULL, county_zoom = NULL, reference_map = FALSE)
```

Arguments

df	A data.frame with a column named "region" and a column named "value". Elements in the "region" column must exactly match how regions are named in the "region" column in county.map.
title	An optional title for the map.
legend	An optional name for the legend.
num_colors	The number of colors to use on the map. A value of 0 uses a divergent scale (useful for visualizing negative and positive numbers), A value of 1 uses a continuous scale (useful for visualizing outliers), and a value in [2, 9] will use that many quantiles.
state_zoom	An optional vector of states to zoom in on. Elements of this vector must exactly match the names of states as they appear in the "region" column of ?state.regions.
county_zoom	An optional vector of counties to zoom in on. Elements of this vector must exactly match the names of counties as they appear in the "region" column of ?county.regions.
reference_map	If true, render the choropleth over a reference map from Google Maps.

Examples

```
## Not run:
# default parameters
data(df_pop_county)
county_choropleth(df_pop_county,
                  title  = "US 2012 County Population Estimates",
                  legend = "Population")

# zoom in on california and add a reference map
county_choropleth(df_pop_county,
                  title         = "California County Population Estimates",
                  legend        = "Population",
                  state_zoom    = "california",
                  reference_map = TRUE)

# continuous scale
data(df_pop_county)
county_choropleth(df_pop_county,
                 title      = "US 2012 County Population Estimates",
                 legend     = "Population",
                 num_colors = 1,
                 state_zoom = c("california", "oregon", "washington"))

library(dplyr)
library(choroplethrMaps)
data(county.regions)

# show the population of the 5 counties (boroughs) that make up New York City
nyc_county_names = c("kings", "bronx", "new york", "queens", "richmond")
nyc_county_fips = county.regions %>%
  filter(state.name == "new york" & county.name %in% nyc_county_names) %>%
```

```
  select(region)
county_choropleth(df_pop_county,
                  title       = "Population of Counties in New York City",
                  legend      = "Population",
                  num_colors  = 1,
                  county_zoom = nyc_county_fips$region)

## End(Not run)
```

`county_choropleth_acs` *Create a US County choropleth from ACS data*

Description

Creates a US County choropleth using the US Census' American Community Survey (ACS) data. Requires the acs package to be installed, and a Census API Key to be set with the acs's api.key.install function. Census API keys can be obtained at http://www.census.gov/developers/tos/key_request.html.

Usage

```
county_choropleth_acs(tableId, endyear = 2011, span = 5, num_colors = 7,
  state_zoom = NULL, county_zoom = NULL)
```

Arguments

`tableId`	The id of an ACS table
`endyear`	The end year of the survey to use. See acs.fetch (?acs.fetch) and http://1.usa.gov/1geFSSj for details.
`span`	The span of time to use. See acs.fetch and http://1.usa.gov/1geFSSj for details.
`num_colors`	The number of colors on the map. A value of 1 will use a continuous scale. A value in [2, 9] will use that many colors.
`state_zoom`	An optional vector of states to zoom in on. Elements of this vector must exactly match the names of states as they appear in the "region" column of ?state.regions.
`county_zoom`	An optional vector of counties to zoom in on. Elements of this vector must exactly match the names of counties as they appear in the "region" column of ?county.regions.

Value

A choropleth.

References

Uses the acs package created by Ezra Haber Glenn.

See Also

`api.key.install` in the acs package which sets an Census API key for the acs library

http://factfinder2.census.gov/faces/help/jsf/pages/metadata.xhtml?lang=en&type=survey&id=survey.en.ACS_ACS which contains a list of all ACS surveys.

Examples

```
## Not run:
# median income, all counties in US
county_choropleth_acs("B19301")

# continuous scale, zooing in on all counties in New York, New Jersey and Connecticut
county_choropleth_acs("B19301", num_colors=1, state_zoom=c("new york", "new jersey", "connecticut"))

# zooming in on the 5 counties (boroughs) that make up New York City
library(dplyr)
library(choroplethrMaps)
data(county.regions)

nyc_county_names=c("kings", "bronx", "new york", "queens", "richmond")
nyc_county_fips = county.regions %>%
  filter(state.name=="new york" & county.name %in% nyc_county_names) %>%
  select(region)
county_choropleth_acs("B19301", num_colors=1, county_zoom=nyc_county_fips$region)

## End(Not run)
```

---

county_zoom_choropleth

Create a choropleth of USA Counties, with sensible defaults, that zooms on counties.

Description

This function is deprecated as of choroplethr version 3.0.0. Please use ?county_choropleth with the county_zoom parameter set instead. The last version of choroplethr in which this function worked was version 2.1.1, which can be downloaded from CRAN here: http://cran.r-project.org/web/packages/choroplethr/index.htm

Usage

```
county_zoom_choropleth(...)
```

Arguments

...	All arguments are ignored.

df_county_demographics *A data.frame containing demographic statistics for each county in the United States.*

Description

A data.frame containing demographic statistics for each county in the United States.

Usage

```
data(df_county_demographics)
```

References

Data comes from the 2013 5-year American Community Survey (ACS). Data generated by ?get_county_demographics.

Examples

```
## Not run:
library(choroplethr)
data(df_county_demographics)

# examine the 2013, 5-year county percent hispanic estimates as a boxplot and choropleth

# the boxplot shows the distribution
boxplot(df_county_demographics$percent_hispanic)

# the choropleth map shows the location of the values
# first set the 'value' column to be the column we want to render
df_county_demographics$value = df_county_demographics$percent_hispanic
county_choropleth(df_county_demographics)

## End(Not run)
```

df_japan_census *A data.frame containing basic demographic information about Japan.*

Description

A data.frame containing basic demographic information about Japan.

Usage

```
data(df_japan_census)
```

References

Taken from the "Total Population" table from the Statistics Bureau of Japan website (http://www.stat.go.jp/english/data/nenkan/1431-02.htm) on 12/1/2014.

`df_pop_country` *A data.frame containing population estimates for Countries in 2012.*

Description

A data.frame containing population estimates for Countries in 2012.

Usage

```
data(df_pop_country)
```

References

Taken from the WDI package with code SP.POP.TOTL for year 2012.

`df_pop_county` *A data.frame containing population estimates for US Counties in 2012.*

Description

A data.frame containing population estimates for US Counties in 2012.

Usage

```
data(df_pop_county)
```

References

Taken from the US American Community Survey (ACS) 5 year estimates.

df_pop_state *A data.frame containing population estimates for US States in 2012.*

Description

A data.frame containing population estimates for US States in 2012.

Usage

```
data(df_pop_state)
```

References

Taken from the US American Community Survey (ACS) 5 year estimates.

df_president *A data.frame containing election results from the 2012 US Presidential election.*

Description

A data.frame containing election results from the 2012 US Presidential election.

Usage

```
data(df_president)
```

Author(s)

Ari Lamstein and Richard Careaga

References

Taken from the FEC website on 11/21/2014.

`df_president_ts` *A data.frame containing all US presdiential election results from 1789 to 2012*

Description

Legend:

- R = Republican
- D = Democratic
- DR = Democratic-Republican
- W = Whig
- F = Federalist
- GW = George Washington
- NR = National Republican
- SD = Southern Democrat
- PR = Progressive
- AI = American Independent
- SR = States' Rights
- PO = Populist
- CU = Constitutional Union
- I = Independent
- ND = Northern Democrat
- KN = Know Nothing
- AM = Anti-Masonic
- N = Nullifier
- SP = Split evenly

Usage

```
data(df_president_ts)
```

References

Taken from http://en.wikipedia.org/wiki/List_of_United_States_presidential_election_results_by_state 3/20/2014.

df_state_age_2010 *A data.frame containing median age estimates for US states in 2010*

Description

A data.frame containing median age estimates for US states in 2010

Usage

```
data(df_state_age_2010)
```

References

Taken from the US American Community Survey (ACS) 5 year estimates.

df_state_age_2015 *A data.frame containing median age estimates for US states in 2015*

Description

A data.frame containing median age estimates for US states in 2015

Usage

```
data(df_state_age_2015)
```

References

Taken from the US American Community Survey (ACS) 5 year estimates.

df_state_demographics *A data.frame containing demographic statistics for each state plus the District of Columbia.*

Description

A data.frame containing demographic statistics for each state plus the District of Columbia.

Usage

```
data(df_state_demographics)
```

References

Data comes from the 2013 5-year American Community Survey (ACS). Data generated by ?get_state_demographics.

Examples

```
## Not run:
library(choroplethr)
data(df_state_demographics)

# examine the 2013, 5-year state percent hispanic estimates as a boxplot and choropleth

# the boxplot shows the distribution
boxplot(df_state_demographics$percent_hispanic)

# the choropleth map shows the location of the values
# first set the 'value' column to be the column we want to render
df_state_demographics$value = df_state_demographics$percent_hispanic
state_choropleth(df_state_demographics)

## End(Not run)
```

`double_map` *Place two maps side by side*

Description

With an optional title. Especially useful for contrasting choropleth maps both with and without a reference map underneath.

Usage

```
double_map(map1, map2, title = "")
```

Arguments

`map1`	The first map
`map2`	The second map
`title`	An optional title

`get_acs_data` *Returns a list representing American Community Survey (ACS) estimates*

Description

Given a map, ACS tableId, endyear and span. Prompts user for the column id if there are multiple tables. The first element of the list is a data.frame with estimates. The second element is the ACS title of the column. Requires the acs package to be installed, and a Census API Key to be set with the acs's api.key.install function. Census API keys can be obtained at http://api.census.gov/data/key_signup.html.

Usage

```
get_acs_data(tableId, map, endyear = 2012, span = 5, column_idx = -1,
  include_moe = FALSE)
```

Arguments

tableId	The id of an ACS table
map	The map you want to use. Must be one of "state", "county" or "zip".
endyear	The end year of the survey to use. See acs.fetch (?acs.fetch) and http://1.usa.gov/1geFSSj for details.
span	The span of time to use. See acs.fetch and http://1.usa.gov/1geFSSj for details. on the same longitude and latitude map to scale. This variable is only checked when the "states" variable is equal to all 50 states.
column_idx	The optional column id of the table to use. If not specified and the table has multiple columns, you will be prompted for a column id.
include_moe	Whether to include the 90 percent margin of error.

See Also

http://factfinder2.census.gov/faces/help/jsf/pages/metadata.xhtml?lang=en&type=survey&id=survey.en.ACS_ACS, which lists all ACS Surveys.

Examples

```
## Not run:
library(Hmisc) # for cut2
# States with greater than 1M residents
df       = get_acs_data("B01003", "state")[[1]] # population
df$value = cut2(df$value, cuts=c(0,1000000,Inf))
state_choropleth(df, title="States with a population over 1M", legend="Population")

# Counties with greater than or greater than 1M residents
df       = get_acs_data("B01003", "county")[[1]] # population
df$value = cut2(df$value, cuts=c(0,1000000,Inf))
county_choropleth(df, title="Counties with a population over 1M", legend="Population")

## End(Not run)
```

---

get_acs_df *Returns a data.frame representing US Census American Community Survey (ACS) estimates.*

Description

This function is deprecated as of choroplethr version 3.0.0. Please use ?get_acs_data instead. The last version of choroplethr in which this functions worked was version 2.1.1, which can be downloaded from CRAN here: http://cran.r-project.org/web/packages/choroplethr/index.html

Usage

```
get_acs_df(...)
```

Arguments

...	All arguments are ignored.

`get_county_demographics` *Get a handful of demographic variables on US Counties from the US Census Bureau as a data.frame.*

Description

The data comes from the American Community Survey (ACS). The variables are: total population, percent White not Hispanic, Percent Black or African American not Hispanic, percent Asian not Hispanic, percent Hispanic all races, per-capita income, median rent and median age.

Usage

```
get_county_demographics(endyear = 2013, span = 5)
```

Arguments

endyear	The end year for the survey
span	The span of the survey

References

The choroplethr guide to Census data: http://www.arilamstein.com/open-source/choroplethr/mapping-us-census-data/

A list of all ACS Surveys: http://factfinder.census.gov/faces/affhelp/jsf/pages/metadata.xhtml?lang=en&type=survey&id=sur

Examples

```
## Not run:
# get some demographic data on US counties from the 2010 5-year ACS
df = get_county_demographics(endyear=2010, span=5)
colnames(df)

# analyze the percent of people who are white not hispanic
# a boxplot shows the distribution
boxplot(df$percent_white)

# a choropleth map shows the location of the values
# set the 'value' column to be the column we want to render
df$value = df$percent_white
county_choropleth(df)
```

```
## End(Not run)
```

`get_state_demographics` *Get a handful of demographic variables on US States from the US Census Bureau as a data.frame.*

Description

The data comes from the American Community Survey (ACS). The variables are: total population, percent White not Hispanic, Percent Black or African American not Hispanic, percent Asian not Hispanic, percent Hispanic all races, per-capita income, median rent and median age.

Usage

```
get_state_demographics(endyear = 2013, span = 5)
```

Arguments

endyear	The end year for the survey
span	The span of the survey

References

The choroplethr guide to Census data: http://www.arilamstein.com/open-source/choroplethr/mapping-us-census-data/

A list of all ACS Surveys: http://factfinder.census.gov/faces/affhelp/jsf/pages/metadata.xhtml?lang=en&type=survey&id=sur

Examples

```
## Not run:
# get some demographic data on US states from the 2010 5-year ACS
df = get_state_demographics(endyear=2010, span=5)
colnames(df)

# analyze the percent of people who are white not hispanic
# a boxplot shows the distribution
boxplot(df$percent_white)

# a choropleth map shows the location of the values
# set the 'value' column to be the column we want to render
df$value = df$percent_white
state_choropleth(df)

## End(Not run)
```

get_tract_demographics

Get a handful of demographic variables on Census Tracts in a State from the US Census Bureau as a data.frame.

Description

The data comes from the American Community Survey (ACS). The variables are: total population, percent White not Hispanic, Percent Black or African American not Hispanic, percent Asian not Hispanic, percent Hispanic all races, per-capita income, median rent and median age.

Usage

```
get_tract_demographics(state_name, county_fips = NULL, endyear = 2013,
  span = 5)
```

Arguments

state_name	The name of the state. See ?state.regions for proper spelling and capitalization.
county_fips	An optional vector of county fips codes within the state. Useful to set because getting data on all tracts can be slow.
endyear	The end year for the survey
span	The span of the survey

References

The choroplethr guide to Census data: http://www.arilamstein.com/open-source/choroplethr/mapping-us-census-data/

A list of all ACS Surveys: http://factfinder.census.gov/faces/affhelp/jsf/pages/metadata.xhtml?lang=en&type=survey&id=sur

get_tract_map

Get a map of tracts in a state, as a data.frame

Description

The map returned is exactly the same map which tract_choropleth uses. It is downloaded using the "tracts" function in the tigris package, and then it is modified for use with choroplethr.

Usage

```
get_tract_map(state_name)
```

Arguments

state_name	The name of the state. See ?state.regions for proper spelling and capitalization.

StateChoropleth *Create a state-level choropleth*

Description

Create a state-level choropleth

Usage

```
StateChoropleth
```

Format

An object of class `R6ClassGenerator` of length 24.

state_choropleth *Create a choropleth of US States*

Description

The map used is state.map in the package choroplethrMaps. See state.regions in the choroplethrMaps package for a data.frame that can help you coerce your regions into the required format.

Usage

```
state_choropleth(df, title = "", legend = "", num_colors = 7,
  zoom = NULL, reference_map = FALSE)
```

Arguments

`df`	A data.frame with a column named "region" and a column named "value". Elements in the "region" column must exactly match how regions are named in the "region" column in state.map.
`title`	An optional title for the map.
`legend`	An optional name for the legend.
`num_colors`	The number of colors to use on the map. A value of 0 uses a divergent scale (useful for visualizing negative and positive numbers), A value of 1 uses a continuous scale (useful for visualizing outliers), and a value in [2, 9] will use that many quantiles.
`zoom`	An optional vector of states to zoom in on. Elements of this vector must exactly match the names of states as they appear in the "region" column of ?state.regions.
`reference_map`	If true, render the choropleth over a reference map from Google Maps.

Examples

```
## Not run:
# default parameters
data(df_pop_state)
state_choropleth(df_pop_state,
                 title  = "US 2012 State Population Estimates",
                 legend = "Population")

# choropleth over reference map of continental usa
data(continental_us_states)
state_choropleth(df_pop_state,
                 title         = "US 2012 State Population Estimates",
                 legend        = "Population",
                 zoom          = continental_us_states,
                 reference_map = TRUE)

# continuous scale and zoom
data(df_pop_state)
state_choropleth(df_pop_state,
                 title      = "US 2012 State Population Estimates",
                 legend     = "Population",
                 num_colors = 1,
                 zoom       = c("california", "oregon", "washington"))

# demonstrate user creating their own discretization of the input
# demonstrate how choroplethr handles character and factor values
data(df_pop_state)
df_pop_state$str = ""
for (i in 1:nrow(df_pop_state))
{
  if (df_pop_state[i,"value"] < 1000000)
  {
    df_pop_state[i,"str"] = "< 1M"
  } else {
    df_pop_state[i,"str"] = "> 1M"
  }
}
df_pop_state$value = df_pop_state$str
state_choropleth(df_pop_state, title = "Which states have less than 1M people?")


## End(Not run)
```

`state_choropleth_acs` *Create a US State choropleth from ACS data*

Description

Creates a choropleth of US States using the US Census' American Community Survey (ACS) data. Requires the acs package to be installed, and a Census API Key to be set with the acs's api.key.install function. Census API keys can be obtained at http://www.census.gov/developers/tos/key_request.html.

Usage

```
state_choropleth_acs(tableId, endyear = 2011, span = 5, num_colors = 7,
  zoom = NULL)
```

Arguments

tableId	The id of an ACS table
endyear	The end year of the survey to use. See acs.fetch (?acs.fetch) and http://1.usa.gov/1geFSSj for details.
span	The span of time to use. See acs.fetch and http://1.usa.gov/1geFSSj for details.
num_colors	The number of colors on the map. A value of 1 will use a continuous scale. A value in [2, 9] will use that many colors.
zoom	An optional list of states to zoom in on. Must come from the "name" column in ?state.regions.

Value

A choropleth.

References

Uses the acs package created by Ezra Haber Glenn.

See Also

`api.key.install` in the acs package which sets an Census API key for the acs library

http://factfinder2.census.gov/faces/help/jsf/pages/metadata.xhtml?lang=en&type=survey&id=survey.en.ACS_ACS which contains a list of all ACS surveys.

Examples

```
## Not run:
# median income, default parameters
state_choropleth_acs("B19301")

# continuous scale, zooming in on New York, New Jersey and Connecticut
state_choropleth_acs("B19301", num_colors=1, zoom=c("new york", "new jersey", "connecticut"))

## End(Not run)
```

TractChoropleth *An R6 object for creating choropleths of Census Tracts.*

Description

An R6 object for creating choropleths of Census Tracts.

Usage

```
TractChoropleth
```

Format

An object of class `R6ClassGenerator` of length 24.

tract_choropleth *Create a choropleth of Census Tracts in a particular state.*

Description

Create a choropleth of Census Tracts in a particular state.

Usage

```
tract_choropleth(df, state_name, title = "", legend = "", num_colors = 7,
  tract_zoom = NULL, county_zoom = NULL, reference_map = FALSE)
```

Arguments

df	A data.frame with a column named "region" and a column named "value".
state_name	The name of the state. See ?state.regions for proper spelling and capitalization.
title	An optional title for the map.
legend	An optional name for the legend.
num_colors	The number of colors to use on the map. A value of 0 uses a divergent scale (useful for visualizing negative and positive numbers), A value of 1 uses a continuous scale (useful for visualizing outliers), and a value in [2, 9] will use that many quantiles.
tract_zoom	An optional vector of tracts to zoom in on. Elements of this vector must exactly match the names of tracts as they appear in the "region" column of the object returned from "get_tract_map".
county_zoom	An optional vector of county FIPS codes to zoom in on. Elements of this vector must exactly match the names of counties as they appear in the "county.fips.numeric" column of the object returned from "get_tract_map".
reference_map	If true, render the choropleth over a reference map from Google Maps.

See Also

https://www.census.gov/geo/reference/gtc/gtc_ct.html for more information on Census Tracts

---

`USAChoropleth` *Normal choropleth that draws Alaska and Hawaii as insets. In addition to a columns named "region" and "value", also requires a column named "state".*

Description

Normal choropleth that draws Alaska and Hawaii as insets. In addition to a columns named "region" and "value", also requires a column named "state".

Usage

```
USAChoropleth
```

Format

An object of class `R6ClassGenerator` of length 24.

`zip_map` *Create a map visualizing US ZIP codes with sensible defaults*

Description

This function is deprecated as of choroplethr version 3.0.0. Please use ?zip_choropleth instead. The last version of choroplethr in which this function worked was version 2.1.1, which can be downloaded from CRAN here: http://cran.r-project.org/web/packages/choroplethr/index.html

Usage

```
zip_map(...)
```

Arguments

...	All arguments are ignored.

Index