Package 'usmap'

August 15, 2017

Version 0.2.1

Title US Maps Including Alaska and Hawaii

Description Obtain United States map data frames of varying region types (e.g. county, state). The map data frames include Alaska and Hawaii conveniently placed to the bottom left, as they appear in most maps of the US. Convenience functions for plotting choropleths and working with FIPS codes are also provided.

Depends R (>= 3.2.0)

License GPL-3 | file LICENSE

Encoding UTF-8

LazyData true

URL https://dilorenzo.pl/usmap

BugReports https://github.com/pdil/usmap/issues

Imports utils

Suggests ggplot2, knitr, proto, rmarkdown, scales, stringr, testthat

RoxygenNote 6.0.1

VignetteBuilder knitr

NeedsCompilation no

Author Paolo Di Lorenzo [aut, cre]

Maintainer Paolo Di Lorenzo <paolo@dilorenzo.pl>

Repository CRAN

Date/Publication 2017-08-15 05:15:17 UTC

R topics documented:

countypop 2
countypov 3
fips 3
fips_info 4
map_with_data 5

plot_usmap 6
statepop 7
statepov 8
usmap 8
us_map 10

Index **11**

`countypop` *Population estimates (2015), county level*

Description

US census population estimates by county for 2015.

The data is formatted for easy merging with output from `us_map`.

Usage

```
data(countypop)
```

Format

A data frame with 3142 rows and 4 variables.

Details

- `fips` The 5-digit FIPS code corresponding to the county.
- `abbr` The 2-letter state abbreviation.
- `county` The full county name.
- `pop_2015` The 2015 population estimate (in number of people) for the corresponding county.

References

- http://www.census.gov/programs-surveys/popest.html
- https://www.ers.usda.gov/data-products/county-level-data-sets/

countypov *Poverty percentage estimates (2014), county level*

Description

US census poverty percentage estimates by county for 2014.

The data is formatted for easy merging with output from `us_map`.

Usage

```
data(countypov)
```

Format

A data frame with 3142 rows and 4 variables.

Details

- `fips` The 5-digit FIPS code corresponding to the county.
- `abbr` The 2-letter state abbreviation.
- `county` The full county name.
- `pct_pov_2014` The 2014 poverty estimate (in percent of county population) for the corresponding county.

References

- https://www.census.gov/topics/income-poverty/poverty.html
- https://www.ers.usda.gov/data-products/county-level-data-sets/

fips *Retrieve FIPS code for either a US state or county*

Description

Each US state and county has a unique FIPS (Federal Information Processing Standards) code. Use this function to obtain the FIPS code for a state or county.

Usage

```
fips(state, county = "")
```

Arguments

state	The state for which to obtain a FIPS code. Can be entered as either a state abbreviation or full name (case-insensitive).
county	The county for which to obtain a FIPS code. Can be entered with or without "county" (case-insensitive).

Details

State and county FIPS (Federal Information Processing Standards) are two and five digit codes, respectively. They uniquely identify all states and counties within the United States. The first two digits of the five digit county codes correspond to the state that the county belongs to. FIPS codes also exist for US territories and minor outlying islands, though this package only provides information for the 50 US states (and their associated counties and census designated areas).

Value

The FIPS code of given `state` or `county`.

Note

A `state` must be included when searching for `county`, otherwise multiple results may be returned for duplicate county names.

Examples

```
fips("NJ")
fips("California")
fips("CA", county = "orange")
fips(state = "AL", county = "autauga")
fips(state = "Alabama", county = "Autauga County")
```

`fips_info` *Retrieve states or counties using FIPS codes*

Description

Retrieve states or counties using FIPS codes

Usage

```
fips_info(fips)

## S3 method for class 'numeric'
fips_info(fips)

## S3 method for class 'character'
fips_info(fips)
```

Arguments

fips	A one to five digit, either numeric or character, vector of FIPS codes for which to look up states or counties. States have a two digit FIPS code and counties have a five digit FIPS code (where the first 2 numbers pertain to the state).

Value

A data frame with the states or counties and the associated FIPS codes.

Examples

```
fips_info(2)
fips_info("2")
fips_info(c("02", "03", "04"))

fips_info(2016)
fips_info(c("02016", "02017"))
```

---

map_with_data *Join county or state level data to US map data*

Description

Join county or state level data to US map data

Usage

```
map_with_data(data, values = "values", include = c(), na = NA)
```

Arguments

data	The data that should be joined to a US map. This parameter should be a data frame consisting of two columns, a fips code (2 characters for state, 5 characters for county) and the value that should be associated with that region. The columns of data *must* be fips and the value of the 'values' parameter.
values	The name of the column that contains the values to be associated with a given region. The default is "values".
include	The regions to include in the output data frame. If regions is "states"/"state", the value can be either a state name, abbreviation or FIPS code. For counties, the FIPS must be provided as there can be multiple counties with the same name.
na	The value to be inserted for states or counties that don't have a value in data. This value must be of the same type as the value column of data.

Value

A data frame composed of the map data frame (from us_map) except an extra column containing the values in data is included.

The result can be plotted using ggplot2. See us_map for more details.

Examples

```
state_data <- data.frame(fips = c("01", "02", "04"), values = c(1, 5, 8))
df <- map_with_data(state_data, na = 0)
```

---

plot_usmap *Conveniently plot basic US map*

Description

Conveniently plot basic US map

Usage

```
plot_usmap(regions = c("states", "state", "counties", "county"),
  include = c(), data = data.frame(), values = "values",
  theme = theme_map(), lines = "black")
```

Arguments

regions	The region breakdown for the map, can be one of ("states", "state", "counties", "county"). The default is "states".
include	The regions to include in the output data frame. If regions is "states"/"state", the value can be either a state name, abbreviation or FIPS code. For counties, the FIPS must be provided as there can be multiple counties with the same name.
data	A data frame containing values to plot on the map. This parameter should be a data frame consisting of two columns, a fips code (2 characters for state, 5 characters for county) and the value that should be associated with that region. The columns of data *must* be fips and the value of the 'values' parameter.
values	The name of the column that contains the values to be associated with a given region. The default is "value".
theme	The theme that should be used for plotting the map. The default is theme_map.
lines	The line color to be used in the map. Corresponds to the colour option in the aes mapping. The default is "black". Click here for more color options.

Value

A ggplot object that contains a basic US map with the described parameters. Since the result is a ggplot object, it can be extended with more geom layers, scales, labels, themes, etc.

See Also

usmap, theme

Examples

```
plot_usmap()
plot_usmap(regions = "states")
plot_usmap(regions = "counties")
plot_usmap(regions = "state")
plot_usmap(regions = "county")

# Output is ggplot object so it can be extended
# with any number of ggplot layers
library(ggplot2)
plot_usmap(include = c("CA", "NV", "ID", "OR", "WA")) +
  labs(title = "Western States")

# Color maps with data
plot_usmap(data = statepop, values = "pop_2015")
```

---

statepop *Population estimates (2015), state level*

Description

US census population estimates by state for 2015.

The data is formatted for easy merging with output from us_map.

Usage

```
data(statepop)
```

Format

A data frame with 51 rows and 4 variables.

Details

- fips The 2-digit FIPS code corresponding to the state.
- abbr The 2-letter state abbreviation.
- full The full state name.
- pop_2015 The 2015 population estimate (in number of people) for the corresponding state.

References

- http://www.census.gov/programs-surveys/popest.html
- https://www.ers.usda.gov/data-products/county-level-data-sets/

statepov *Poverty percentage estimates (2014), state level*

Description

US census poverty percentage estimates by state for 2014.

The data is formatted for easy merging with output from us_map.

Usage

```
data(statepov)
```

Format

A data frame with 51 rows and 4 variables.

Details

- fips The 2-digit FIPS code corresponding to the state.
- abbr The 2-letter state abbreviation.
- full The full state name.
- pct_pov_2014 The 2014 poverty estimate (in percent of state population) for the corresponding state

References

- https://www.census.gov/topics/income-poverty/poverty.html
- https://www.ers.usda.gov/data-products/county-level-data-sets/

usmap *usmap: US maps including Alaska and Hawaii*

Description

It is usually difficult or inconvenient to create US maps that include both Alaska and Hawaii in a convenient spot. All map data frames produced by this package use the Alber's Equal Area projection.

Map data frames

Alaska and Hawaii have been manually moved to a new location so that their new coordinates place them to the bottom-left corner of the map. These maps can be accessed by using the `usmap` function.

The function provides the ability to retrieve maps with either state borders or county borders using the `regions` parameter for convenience.

States (or counties) can be included such that all other states (or counties) are excluded using the `include` parameter.

FIPS lookup tools

Several functions have been included to lookup the US state or county pertaining to a FIPS code.

Likewise a reverse lookup can be done where a FIPS code can be used to retrieve the associated state(s) or county(ies). This can be useful when preparing data to be merged with the map data frame.

Plot US map data

A convenience function `plot_usmap` has been included which takes similar parameters to `us_map` and returns a **ggplot2** object. Since the output is a `ggplot` object, other layers can be added such as scales, themes, and labels. Including data in the function call will color the map according to the values in the data, creating a choropleth.

Author(s)

Paolo Di Lorenzo

- Email: <paolo@dilorenzo.pl>
- Website: http://dilorenzo.pl
- Twitter: http://twitter.com/dilorenzopl
- Github: http://www.github.com/pdil/

References

Rudis, Bob. "Moving The Earth (well, Alaska & Hawaii) With R." Blog post. Rud.is., 16 Nov. 2014. Web. 10 Aug. 2015. <https://rud.is/b/2014/11/16/moving-the-earth-well-alaska-hawaii-with-r/>.

See Also

Helpful links:

- FIPS code information http://en.wikipedia.org/wiki/FIPS_county_code http://en.wikipedia.org/wiki/FIPS_state_code
- US Census Shapefiles https://www.census.gov/geo/maps-data/data/tiger-cart-boundary.html

- Map Features
https://en.wikipedia.org/wiki/Map_projection https://en.wikipedia.org/wiki/Albers_projection https://en.wikipedia.org/wiki/Choropleth

---

`us_map` *Retrieve US map data*

Description

Retrieve US map data

Usage

```
us_map(regions = c("states", "state", "counties", "county"), include = c())
```

Arguments

`regions`	The region breakdown for the map, can be one of (`"states"`, `"state"`, `"counties"`, `"county"`). The default is `"states"`.
`include`	The regions to include in the output data frame. If `regions` is `"states"`/`"state"`, the value can be either a state name, abbreviation or FIPS code. For counties, the FIPS must be provided as there can be multiple counties with the same name.

Value

A data frame of US map coordinates divided by the desired `regions`.

Examples

```
str(us_map())

df <- us_map(regions = "counties")
west_coast <- us_map(include = c("CA", "OR", "WA"))
```

Index

∗Topic **data**
countypop, 2
countypov, 3
statepop, 7
statepov, 8

aes, *6*

countypop, 2
countypov, 3

fips, 3
fips_info, 4

ggplot, *6*

map_with_data, 5

plot_usmap, 6, *9*

statepop, 7
statepov, 8

theme, *7*
theme_map, *6*

us_map, *2, 3*, *6–9*, 10
usmap, *7*, 8, *9*
usmap-package *(*usmap*)*, 8