Package ‘usmap’

January 21, 2021

Version 0.5.2
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 (>= 3.5.0)
License GPL-3 | file LICENSE
Encoding UTF-8
LazyData true
URL https://usmap.dev
BugReports https://github.com/pdil/usmap/issues
Imports utils
Suggests covr, ggplot2, ggrepel, knitr, maptools, proto, rgdal, rmarkdown, scales, sp, stringr, testthat
RoxygenNote 7.1.1
VignetteBuilder knitr
NeedsCompilation no
Author Paolo Di Lorenzo [aut, cre]
Maintainer Paolo Di Lorenzo <paolo@dilorenzo.pl>
Repository CRAN
Date/Publication 2021-01-21 16:20:03 UTC

R topics documented:

  .east_north_central .................. 2
  .east_south_central .................. 3
  .midwest_region .................... 3
  .mid_atlantic ....................... 4
Description

US Census Bureau regional division containing Illinois, Indiana, Michigan, Ohio, and Wisconsin.

Usage

`.east_north_central`

Format

An object of class `character` of length 5.

Details

See [https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf](https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf)

Examples

```r
plot_usmap(include = .east_north_central, labels = TRUE)
```
.east_south_central  East South Central census division

Description
US Census Bureau regional division containing Alabama, Kentucky, Mississippi, and Tennessee.

Usage
.east_south_central

Format
An object of class character of length 4.

Details
See https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf

Examples
plot_usmap(include = .east_south_central, labels = TRUE)

.midwest_region  Midwest census region

Description
US Census Bureau region containing the East North Central and West North Central divisions. This region was designated as "North Central Region" prior to June 1984.

Usage
.midwest_region

Format
An object of class character of length 12.

Details
See https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf

Examples
plot_usmap(include = .midwest_region, labels = TRUE)
**.mid_atlantic**  
*Mid-Atlantic census division*

**Description**  

**Usage**  
`.mid_atlantic`

**Format**  
An object of class character of length 3.

**Details**  
See [https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf](https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf)

**Examples**  
```r  
plot_usmap(include = .mid_atlantic, labels = TRUE)  
```

**.mountain**  
*Mountain census division*

**Description**  
US Census Bureau regional division containing Arizona, Colorado, Idaho, Montana, Nevada, New Mexico, Utah, and Wyoming.

**Usage**  
`.mountain`

**Format**  
An object of class character of length 8.

**Details**  
See [https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf](https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf)

**Examples**  
```r  
plot_usmap(include = .mountain, labels = TRUE)  
```
.new_england

---

.new_england New England census division

Description

US Census Bureau regional division containing Connecticut, Maine, Massachusetts, New Hampshire, Rhode Island, and Vermont.

Usage

.new_england

Format

An object of class character of length 6.

Details

See https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf

Examples

plot_usmap(include = .new_england, labels = TRUE)

---

.northeast_region Northeast census region

Description

US Census Bureau region containing the New England and Mid-Atlantic divisions.

Usage

.northeast_region

Format

An object of class character of length 9.

Details

See https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf

Examples

plot_usmap(include = .northeast_region, labels = TRUE)
north_central_region  North-Central census region

Description
Former US Census Bureau region containing the East North Central and West North Central divisions. This region has been designated as "Midwest" since June 1984.

Usage
north_central_region

Format
An object of class character of length 12.

Details
See https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf

Examples
plot_usmap(include = .north_central_region, labels = TRUE)

pacific  Pacific census division

Description
US Census Bureau regional division containing Alaska, California, Hawaii, Oregon, and Washington.

Usage
pacific

Format
An object of class character of length 5.

Details
See https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf

Examples
plot_usmap(include = .pacific, labels = TRUE)
### south_atlantic

**Description**

US Census Bureau regional division containing Delaware, Florida, Georgia, Maryland, North Carolina, South Carolina, Virginia, District of Columbia, and West Virginia.

**Usage**

`south_atlantic`

**Format**

An object of class character of length 9.

**Details**

See [https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf](https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf)

**Examples**

```r
plot_usmap(include = south_atlantic, labels = TRUE)
```

### south_region

**Description**

US Census Bureau region containing the South Atlantic, East South Central, and West South Central divisions.

**Usage**

`south_region`

**Format**

An object of class character of length 17.

**Details**

See [https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf](https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf)

**Examples**

```r
plot_usmap(include = midwest_region, labels = TRUE)
```
### west_north_central

**West North Central census division**

**Description**

US Census Bureau regional division containing Iowa, Kansas, Minnesota, Missouri, Nebraska, North Dakota, and South Dakota.

**Usage**

\[\text{.west_north_central}\]

**Format**

An object of class character of length 7.

**Details**

See [https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf](https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf)

**Examples**

\[\text{plot_usmap(include = .west_north_central, labels = TRUE)}\]

---

### west_region

**West census region**

**Description**

US Census Bureau region containing the Mountain and Pacific divisions.

**Usage**

\[\text{.west_region}\]

**Format**

An object of class character of length 13.

**Details**

See [https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf](https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf)

**Examples**

\[\text{plot_usmap(include = .midwest_region, labels = TRUE)}\]
.west_south_central

West South Central census division

Description
US Census Bureau regional division containing Arkansas, Louisiana, Oklahoma, and Texas.

Usage
.west_south_central

Format
An object of class character of length 4.

Details
See https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf

Examples
plot_usmap(include = .west_south_central, labels = TRUE)

citypop

Most populous city in each state (2010)

Description
The most populous city in each US state, as of the 2010 US Census.

The data is formatted for transforming with usmap_transform. Once the longitude and latitude is transformed, it can be added to plot_usmap using ggplot2 layers.

Usage
data(citypop)

Format
A data frame with 51 rows and 5 variables.

Details
- lon The longitude of the most populous city.
- lat The latitude of the most populous city.
- state The name of the state containing the city.
- most_populous_city The name of the city.
- city_pop The population of the city.
References


**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**

```r
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**

- https://www.census.gov/programs-surveys/popest.html

**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**

```r
data(countypov)
```
**earthquakes**

**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.census.gov/topics/income-poverty/poverty.html)

**Description**
US earthquakes with a magnitude of 2.5 or greater, occurring in the first half of 2019, from January 1st to June 30th, from USGS.

The data is formatted for transforming with usmap_transform. Once the longitude and latitude is transformed, it can be added to plot_usmap using ggplot2 layers.

**Usage**
```r
data(earthquakes)
```

**Format**
A data frame with 2254 rows and 3 variables.

**Details**
- **lon** The longitude of the earthquake’s location.
- **lat** The latitude of the earthquake’s location.
- **mag** The magnitude of the earthquake.

**References**
- [https://earthquake.usgs.gov/earthquakes/search/](https://earthquake.usgs.gov/earthquakes/search/)
fips

Retrieves 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 = c())

Arguments

state The state(s) for which to obtain a FIPS code(s). Can be entered as either a state abbreviation or full name (case-insensitive).

‘state’ can be entered as either a single state or a vector of states. If ‘state’ is a vector, ‘county’ must be omitted.

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(s) of given state or county.

If only states are entered, a vector of length equal to the number of states is returned. If any states are not found or are invalid, ‘NA’ is returned in their place.

If a state and county are entered, a single value with the FIPS code for the given county is returned. If the county is invalid for the given state, an error is thrown.

If both ‘state’ and ‘county’ are omitted, the entire list of available FIPS codes is returned, sorted by the state’s abbreviation (e.g. Alaska (AK) comes before Alabama (AL)).

Note

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

fips()

fips("NJ")
fips("California")

fips(c("AK", "CA", "UT"))

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.

If ‘fips’ is omitted, the data frame containing all available states is returned.

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(), exclude = 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 or state and the value of the 'values' parameter. If both fips and state are provided, this function uses the fips.

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 resulting map. 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. If states are provided in the county map, only counties in the included states will be returned.

exclude The regions to exclude in the resulting map. 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. The regions listed in the include parameter are applied first and the exclude regions are then removed from the resulting map. Any excluded regions not present in the included regions will be ignored.

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 or plot_usmap 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)

state_data <- data.frame(state = c("AK", "CA", "Utah"), values = c(6, 9, 3))
```r
df <- map_with_data(state_data, na = 0)
```

---

**Description**

Conveniently plot basic US map

**Usage**

```r
plot_usmap(
  regions = c("states", "state", "counties", "county"),
  include = c(),
  exclude = c(),
  data = data.frame(),
  values = "values",
  theme = theme_map(),
  labels = FALSE,
  label_color = "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 resulting map. 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. If states are provided in the county map, only counties in the included states will be returned.
- `exclude` The regions to exclude in the resulting map. 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. The regions listed in the include parameter are applied first and the exclude regions are then removed from the resulting map. Any excluded regions not present in the included regions will be ignored.
- `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 or state 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 from ggthemes.

labels Whether or not to display labels on the map. Labels are not displayed by default. For now, labels only work for state maps. County labels may be added in the future.

label_color The color of the labels to display. Corresponds to the color option in the aes mapping. The default is "black". Click here for more color options.

... Other arguments to pass to ggplot2::aes(). These are often aesthetics, used to set an aesthetic to a fixed value, like color = "red" or size = 3. They affect the appearance of the polygons used to render the map (for example fill color, line color, line thickness, etc.). If any of color/colour, fill, or size are not specified they are set to their default values of color="black", fill="white", and size=0.4.

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"); label(title = "Western States")

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

# Include labels on map (e.g. state abbreviations)
plot_usmap(data = statepop, values = "pop_2015", labels = TRUE)
# Choose color for labels
plot_usmap(data = statepop, values = "pop_2015", labels = TRUE, label_color = "white")
**Description**

US census population estimates by state for 2015.

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

**Usage**

```r
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**

- [https://www.census.gov/programs-surveys/popest.html](https://www.census.gov/programs-surveys/popest.html)
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.census.gov/topics/income-poverty/poverty.html)

---

**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 Albers 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 `us_map` 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.
**usmap_crs**

**Author(s)**

Paolo Di Lorenzo

- Email: <paolo@dilorenzo.pl>
- Website: https://dilorenzo.pl
- Github: https://github.com/pdil/

**References**


**See Also**

Helpful links:

- FIPS code information
- US Census Shapefiles
- Map Features

---

**Description**

This coordinate reference system (CRS) represents the canonical projection used by the usmap package. It can be used to transform shape files, spatial points, spatial data frames, etc. to the same coordinate representation that is used by the plot_usmap function.

**Usage**

usmap_crs()
Convert coordinate data frame to usmap projection

Description

Converting an external data frame of map coordinates will allow those points to line up with the regular usmap plot by applying the same Albers Equal Area projection to those points as well.

Usage

usmap_transform(data)

## S3 method for class 'data.frame'
usmap_transform(data)

Arguments

data

A data frame containing coordinates in a two column format where the first column represents longitude and the second data frame represents latitude. The names of the data frame column do not matter, just that the order of the columns is kept intact.

Value

A data frame containing the transformed coordinates from the input data frame with the Albers Equal Area projection applied. The transformed columns will be appended to the data frame so that all original columns should remain intact.

Examples

data <- data.frame(
  lon = c(-74.01, -95.36, -118.24, -87.65, -134.42, -157.86),
  lat = c(40.71, 29.76, 34.05, 41.85, 58.30, 21.31),
  pop = c(8398748, 2325502, 3990456, 2705994, 32113, 347397)
)

# Transform data
transformed_data <- usmap_transform(data)

# Plot transformed data on map
library(ggplot2)

plot_usmap() + geom_point(
  data = transformed_data,
  aes(x = lon.1, y = lat.1, size = pop),
  color = "red", alpha = 0.5
)
**us_map**

Retrieve US map data

**Description**

Retrieve US map data

**Usage**

```
us_map(
  regions = c("states", "state", "counties", "county"),
  include = c(),
  exclude = 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 resulting map. 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. If states are provided in the county map, only counties in the included states will be returned.

- **exclude**
  The regions to exclude in the resulting map. 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. The regions listed in the `include` parameter are applied first and the `exclude` regions are then removed from the resulting map. Any excluded regions not present in the included regions will be ignored.

**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"))

south_atl_excl_FL <- us_map(include = .south_atlantic, exclude = "FL")
```
Index

* datasets
  .east_north_central, 2
  .east_south_central, 3
  .mid_atlantic, 4
  .midwest_region, 3
  .mountain, 4
  .new_england, 5
  .north_central_region, 6
  .northeast_region, 5
  .pacific, 6
  .south_atlantic, 7
  .south_region, 7
  .west_north_central, 8
  .west_region, 8
  .west_south_central, 9

* data
  citypop, 9
  countypop, 10
countypov, 10
earthquakes, 11
fips, 12
fips_info, 13
ggplot, 16
map_with_data, 14
plot_usmap, 9, 11, 14, 15, 18
statepop, 17
statepov, 17
tHEME, 16
us_map, 10, 14, 17, 18, 21
usmap, 16, 18
usmap-package (usmap), 18
usmap_crs, 19
usmap_transform, 9, 11, 20

aes, 16
citypop, 9