Package 'usmap'

May 28, 2025

Version 0.8.0

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, visualizing spatial data, and working with FIPS codes are also provided.

Depends R (>= 3.5.0)

License GPL (>= 3)

Encoding UTF-8

Language en-US

LazyData true

URL <https://usmap.dev>

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

Imports rlang, usmapdata (>= 0.4.0)

Suggests covr, ggplot2 (>= 3.5.0), ggrepel, knitr, proto, rmarkdown, scales, sf, spelling, testthat (>= 3.0.0), vdiffr

RoxygenNote 7.3.2

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Paolo Di Lorenzo [aut, cph, cre]

Maintainer Paolo Di Lorenzo <paolo@dilorenzo.org>

Repository CRAN

Date/Publication 2025-05-28 08:00:02 UTC

Contents

.east_north_central   *East North Central census division*

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

Examples

```
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

US Census Bureau regional division containing New Jersey, New York, and Pennsylvania.

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

Examples

```
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

Examples

```
plot_usmap(include = .mountain, labels = TRUE)
```

.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   *South Atlantic census division*

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

Examples

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

.south_region   *South census 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

Examples

```
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

```
.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

Examples

```
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

```
.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

Examples

```
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::ggplot() 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.
- abbr The abbreviation of the state containing the city.
- most_populous_city The name of the city.
- city_pop The population of the city.

References

- https://www.census.gov/programs-surveys/decennial-census/decade.2010.html

---

countypop   *Population estimates (2022), county level*

Description

US census population estimates by county for 2022.

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

Usage

```
data(countypop)
```

Format

A data frame with 3222 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_2022 The 2022 population estimate (in number of people) for the corresponding county.

References

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

---

countypov	*Poverty percentage estimates (2021), county level*

Description

US census poverty percentage estimates by county for 2021.

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

Usage

```
data(countypov)
```

Format

A data frame with 3194 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_2021 The 2021 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

---

earthquakes	*Earthquakes (2019)*

Description

US earthquakes with a magnitude of 2.5 or greater, occurring in the first half of 2019, from January 1 to June 30, 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::ggplot() layers.

Usage

```
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://www.usgs.gov/programs/earthquake-hazards/earthquakes>
- <https://earthquake.usgs.gov/earthquakes/search/>

---

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 = c(), data_year = NULL)
```

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).

data_year
:   The year for which to obtain FIPS data. If the value is NULL, the most recent year's data is used. If the provided year is not found from the available map data sets, the next most recent year's data is used. The default is NULL, i.e. the most recent available year is used.

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.

See Also

[fips_info()](fips_info)

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, sortAndRemoveDuplicates = FALSE, data_year = NULL)

## S3 method for class 'numeric'
fips_info(fips, sortAndRemoveDuplicates = FALSE, data_year = NULL)

## S3 method for class 'character'
fips_info(fips, sortAndRemoveDuplicates = FALSE, data_year = NULL)
```

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).
`sortAndRemoveDuplicates`	
	Whether or not to sort the output and remove duplicates. By default, the output will be returned in the order of the values provided to the `fips` parameter. Set this parameter to `TRUE` to return the output sorted by FIPS with a single instance of each FIPS.
`data_year`	The year for which to obtain FIPS data. If the value is NULL, the most recent year's data is used. If the provided year is not found from the available map data sets, the next most recent year's data is used. The default is NULL, i.e. the most recent available year is used.

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.

See Also

fips()

Examples

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

fips_info(2016)
fips_info(c("02016", "02017"), sortAndRemoveDuplicates = TRUE)
```

---

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,
    data_year = NULL
)
```

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`.
`data_year`	The year for which to obtain map data. If the value is NULL, the most recent year's data is used. If the provided year is not found from the available map data sets, the next most recent year's data is used. This can be used if an older data set is being plotted on the US map so that the data matches the map more accurately. Therefore, the provided value should match the year of the plotted data set. The default is NULL, i.e. the most recent available year is used.

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::ggplot()](#) or [plot_usmap()](#).

See Also

[plot_usmap()](#)

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))
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(),
  exclude = c(),
  data = data.frame(),
  values = "values",
  theme = theme_map(),
  labels = FALSE,
  label_color = "black",
  data_year = NULL,
  ...
)
```

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 pa-
               rameter.

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.
label_color	The color of the labels to display. Corresponds to the color option in the ggplot2::aes() mapping. The default is "black". Click here for more color options.
data_year	The year for which to obtain map data. If the value is NULL, the most recent year's data is used. If the provided year is not found from the available map data sets, the next most recent year's data is used. This can be used if an older data set is being plotted on the US map so that the data matches the map more accurately. Therefore, the provided value should match the year of the plotted data set. The default is NULL, i.e. the most recent available year is used.
...	Other arguments to pass to ggplot2::aes(). These are often aesthetics, used to set an aesthetic to a fixed value, like color = "red" or linewidth = 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 linewidth are not specified they are set to their default values of color="black", fill="white", and linewidth=0.4.

Value

A ggplot2::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 ggplot2::Geom layers, scales, labels, themes, etc.

See Also

usmap, ggplot2::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_2022")

# Include labels on map (e.g. state abbreviations)
```

```
plot_usmap(data = statepop, values = "pop_2022", labels = TRUE)
# Choose color for labels
plot_usmap(data = statepop, values = "pop_2022", labels = TRUE, label_color = "white")
```

---

statepop   *Population estimates (2022), state level*

Description

US census population estimates by state for 2022.

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

Usage

```
data(statepop)
```

Format

A data frame with 52 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_2022` The 2022 population estimate (in number of people) for the corresponding state.

References

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

- https://www.ers.usda.gov/data-products/county-level-data-sets

---

statepov *Poverty percentage estimates (2021), state level*

Description

US census poverty percentage estimates by state for 2021.

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_2021` The 2021 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 presented in this package uses the US National Atlas Equal Area projection.

Map data

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 and excluded using the provided include and exclude parameters. These parameters can be used together with any combination of names, abbreviations, or FIPS code to create more complex maps.

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 states or counties. 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::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.

Transforming data

It is also possible to add spatial data to the map, in the form of either data frames or simple features (sf::sf) objects. If necessary, the data can be transformed to be in the same coordinate reference system as usmap by using usmap_transform() and then plotted using ggplot2::geom_sf().

Author(s)

Paolo Di Lorenzo

- Email: <dilorenzo@hey>
- GitHub: https://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
  https://en.wikipedia.org/wiki/FIPS_county_code https://en.wikipedia.org/wiki/
  FIPS_state_code

- US Census Shapefiles
  https://www.census.gov/geographies/mapping-files/time-series/geo/cartographic-boundary.
  html
- Map Features
  https://en.wikipedia.org/wiki/Map_projection
  https://en.wikipedia.org/wiki/Equal-area_projection
  https://en.wikipedia.org/wiki/Choropleth
  https://epsg.io/9311 (US National Atlas Equal Area)

---

usmap_crs	*usmap coordinate reference system*

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()
```

---

usmap_transform	*Convert spatial data to usmap projection*

Description

Converting a spatial object of map coordinates will allow those points to line up with the regular usmap plot by applying the same US National Atlas Equal Area projection (including Alaska and Hawaii of course) to those points as well.

The input `data` is assumed to contain longitude and latitude coordinates by default. If this is not the case, provide an sf::st_crs object to the `crs` parameter with the appropriate coordinate reference system.

Usage

```
usmap_transform(data, ...)

## S3 method for class 'sf'
usmap_transform(data, ...)

## S3 method for class 'data.frame'
usmap_transform(data, ..., input_names = c("lon", "lat"), output_names = NULL)
```

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.
`...`	Additional parameters passed onto [sf::st_as_sf](). By default, `crs = sf::st_crs(4326)` is used, implying longitude and latitude coordinates.
`input_names`	A character vector of length two which specifies the longitude and latitude columns of the input data (the ones that should be transformed), respectively. Only required if the input data is a `data.frame` object. Defaults to `c("lon", "lat")`.
`output_names`	Defunct, this parameter is no longer used. The output of this function will have a column named `"geometry"` with the transformed coordinates. This parameter may be removed in a future version.

Value

An `sf` object containing the transformed coordinates from the input data frame with the US National Atlas 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_sf(
  data = transformed_data,
  aes(size = pop),
  color = "red", alpha = 0.5
)
```

---

usrivers   *US Major Rivers (2010)*

Description

Major rivers in the United States.

The data is can be transformed with usmap_transform(). Once the Shape strings are transformed, it can be added to plot_usmap() using a ggplot2::geom_sf() layer.

Usage

```
data(usrivers)
```

Format

A simple features (sf) data frame with 55 rows and 5 variables.

Details

- NAME The name of the river.
- SYSTEM The system the river belongs to.
- MILES The length of the river in miles.
- Shape_Length The length of the river in the coordinate system.
- Shape The MULTILINESTRING features depicting the river, for plotting.

References

- https://www.arcgis.com/home/item.html?id=290e4ab8a07f4d2c8392848d011add32#!
- Sources: Esri; Rand McNally; Bartholemew and Times Books; Digital Chart of the World (DCW), U.S. National Geospatial-Intelligence Agency (NGA); i-cubed

---

us_map   *Retrieve US map data*

Description

Retrieve US map data

Usage

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

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_year	The year for which to obtain map data. If the value is NULL, the most recent year's data is used. If the provided year is not found from the available map data sets, the next most recent year's data is used. This can be used if an older data set is being plotted on the US map so that the data matches the map more accurately. Therefore, the provided value should match the year of the plotted data set. The default is NULL, i.e. the most recent available year is used.

Value

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

See Also

usmapdata::us_map() of which this function is a wrapper for.

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