Package: alarmdata (via r-universe)

October 10, 2024

Title Download, Merge, and Process Redistricting Data

Version 0.2.2

Description Utility functions to download and process data produced by the ALARM Project, including 2020 redistricting files Kenny and McCartan (2021) <https://alarm-redist.org/posts/2021-08-10-census-2020/> and the 50-State Redistricting Simulations of McCartan, Kenny, Simko, Garcia, Wang, Wu, Kuriwaki, and Imai (2022) <doi:10.7910/DVN/SLCD3E>. The package extends the data introduced in McCartan, Kenny, Simko, Garcia, Wang, Wu, Kuriwaki, and Imai (2022) <doi:10.1038/s41597-022-01808-2> to also include states with only a single district.

Depends R (>= 3.10)

Imports rlang, cli, curl, dplyr, readr, stringr, sf, dataverse, censable, geomander (>= 2.1.0), tidyselect, redist (>= 4.2.0), redistmetrics, tinytiger, rappdirs

Suggests rstudioapi, rmapshaper, testthat (>= 3.0.0), spelling

License MIT + file LICENSE

URL https://github.com/alarm-redist/alarmdata/, https://alarm-redist.org/alarmdata/

BugReports https://github.com/alarm-redist/alarmdata/issues/

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Language en-US

Repository https://christopherkenny.r-universe.dev

RemoteUrl https://github.com/alarm-redist/alarmdata

RemoteRef HEAD

RemoteSha 80a6e115f386a07e6839b99f2f8f0dcd4de98c29

Contents

alarm_50state *Download maps and plans from the 50-State Simulation Project*

Description

These functions will download redist_map and redist_plans objects for the 50-State Simulation Project from the ALARM Project's Dataverse. `alarm_50state_doc()` will download documentation for a particular state and show it in a browser. `alarm_50state_stats` will download just the summary statistics for a state.

Usage

```
alarm_50state_map(state, year = 2020, refresh = FALSE)

alarm_50state_plans(
  state,
  stats = TRUE,
  year = 2020,
  refresh = FALSE,
  compress = "xz"
)

alarm_50state_stats(state, year = 2020, refresh = FALSE)

alarm_50state_doc(state, year = 2020)
```

Arguments

`state`	A state name, abbreviation, FIPS code, or ANSI code.
`year`	The redistricting cycle to download. Currently only `2020` and `2010` are available.
`refresh`	If `TRUE`, ignore the cache and download again.
`stats`	If `TRUE` (the default), download summary statistics for each plan.
`compress`	The compression level used for caching redist_plans objects.

Details

Every decade following the Census, states and municipalities must redraw districts for Congress, state houses, city councils, and more. The goal of the 50-State Simulation Project is to enable researchers, practitioners, and the general public to use cutting-edge redistricting simulation analysis to evaluate enacted congressional districts.

Evaluating a redistricting plan requires analysts to take into account each state's redistricting rules and particular political geography. Comparing the partisan bias of a plan for Texas with the bias of a plan for New York, for example, is likely misleading. Comparing a state's current plan to a past plan is also problematic because of demographic and political changes over time. Redistricting simulations generate an ensemble of alternative redistricting plans within a given state which are tailored to its redistricting rules. Unlike traditional evaluation methods, therefore, simulations are able to directly account for the state's political geography and redistricting criteria.

Value

For `alarm_50state_map()`, a redist_map. For `alarm_50state_plans()`, a redist_plans. For `alarm_50state_doc()`, invisibly returns the path to the HTML documentation, and also loads an HTML file into the viewer or web browser. For `alarm_50state_stats()`, a tibble.

Examples

```
# requires Harvard Dataverse API key
alarm_50state_map("WA")
alarm_50state_plans("WA", stats = FALSE)
alarm_50state_stats("WA")
alarm_50state_doc("WA")

map <- alarm_50state_map("WY")
pl <- alarm_50state_plans("WY")
```

alarm_add_plan *Add a reference plan to a set of plans*

Description

Facilitates comparing an existing (i.e., non-simulated) redistricting plan to a set of simulated plans.

Usage

```
alarm_add_plan(
  plans,
  ref_plan,
  map = NULL,
  name = NULL,
  calc_polsby = FALSE,
  GEOID = "GEOID",
  year = 2020
)
```

Arguments

`plans` A `redist_plans` object.

`ref_plan` An `integer` vector containing the reference plan, a block assignment file as a `tibble` or `data.frame`, or an `sf` object where each row corresponds to a district.

`map` A `redist_map` object. Only required if the `redist_plans` object includes summary statistics.

`name` A human-readable name for the reference plan. Defaults to the name of `ref_plan`. If `ref_plan` is a `tibble` or `data.frame`, it should be the name of the column of `ref_plan` that identifies districts.

`calc_polsby` A logical value indicating whether a Polsby-Popper compactness score should be calculated for the reference plan. Defaults to `FALSE`.

`GEOID` character. If `ref_plan` is a `tibble` or `data.frame`, then it should correspond to the column of `ref_plan` that identifies block `GEOID`s. If `ref_plan` is an `sf` object, then it should correspond to the column of `ref_plan` that identifies district numbers. Ignored when `ref_plan` is numeric. Default is `'GEOID'`.

`year` the decade to request if passing a `tibble` to `ref_plan`, either `2010` or `2020`. Default is `2020`.

Value

A modified `redist_plans` object containing the reference plan. Includes summary statistics if the original `redist_plans` object had them as well.

Examples

```
# requires Harvard Dataverse API key
map <- alarm_50state_map("WY")
pl <- alarm_50state_plans("WY")
pl_new <- alarm_add_plan(pl, ref_plan = c(1), map, name = "example")

# download and load a comparison plan
url <- paste0("https://github.com/PlanScore/Redistrict2020/raw/main/files/",
  "NM-2021-10/Congressional_Concept_A.zip")
tf <- tempfile(fileext = ".zip")
utils::download.file(url, tf)
utils::unzip(tf, exdir = dirname(tf))
baf <- readr::read_csv(file = paste0(dirname(tf), "/Congressional Concept A.csv"),
                       col_types = "ci")
names(baf) <- c("GEOID", "concept_a")
# Add it to the plans object
map_nm <- alarm_50state_map("NM")
plans_nm <- alarm_50state_plans("NM", stats = FALSE)
alarm_add_plan(plans_nm, baf, map = map_nm, name = "concept_a")
```

alarm_cache_size *Work with the the* alarmdata *cache*

Description

Functions to inspect and clear the cache. If the cache is not enabled, uses a temporary directory.

Usage

```
alarm_cache_size()

alarm_cache_clear(force = FALSE)

alarm_cache_path()
```

Arguments

force	FALSE by default. Asks the user to confirm if interactive. Does not clear cache if force is FALSE and not interactive.

Value

For `alarm_cache_size()`, the size in bytes, invisibly

For `alarm_cache_clear()`, the path to the cache, invisibly.

For `alarm_cache_path()`, the path to the cache

Examples

```
alarm_cache_size()

alarm_cache_clear()

alarm_cache_path()
```

alarm_census_vest *Download Joined VEST and Census Data*

Description

Downloads Census data joined with VEST's election data. All are re-tabulated from precincts collected by VEST to 2020 Census geographies.

Usage

```
alarm_census_vest(state, geometry = FALSE, epsg = alarm_epsg(state))
```

Arguments

`state` A state name, abbreviation, FIPS code, or ANSI code.

`geometry` If `TRUE` (default is `FALSE`), include `sf` geometry from Census Bureau TIGER Lines with the data.

`epsg` A numeric EPSG code to use as the coordinate system. Default is `alarm_epsg(state)`.

Value

tibble with Census and election data

Examples

```
alarm_census_vest("DE", geometry = FALSE)
```

`alarm_epsg` *Suggested EPSG Codes*

Description

Provides suggested EPSG codes for each of the 50 states. One of the NAD83 (HARN) coordinate systems for each state.

Usage

```
alarm_epsg(state)
```

Arguments

`state` A state name, abbreviation, FIPS code, or ANSI code.

Value

A numeric EPSG code

Examples

```
alarm_epsg("NY")
```

Index