Package ‘easycensus’

February 28, 2022

Title Quickly Extract and Marginalize U.S. Census Tables

Version 0.2.1

Description Extracting desired data using the proper Census variable names can be time-consuming. This package takes the pain out of that process by providing functions to quickly locate variables and download labeled tables from the Census APIs (<https://www.census.gov/data/developers/data-sets.html>).

Depends R (>= 2.10)

Imports rlang, dplyr (>= 1.0.0), stringr, tidycensus, cli

Suggests testthat (>= 3.0.0), sf, purrr

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.2

URL https://corymccartan.github.io/easycensus/

BugReports https://github.com/CoryMcCartan/easycensus/issues

Config/testthat/edition 3

Language en-US

NeedsCompilation no

Author Cory McCartan [aut, cre]

Maintainer Cory McCartan <cmccartan@g.harvard.edu>

Repository CRAN

Date/Publication 2022-02-28 19:20:02 UTC

R topics documented:

find_table *Find a decennial or ACS census table with variables of interest*

Description

This function uses fuzzy matching to help identify tables from the census which contain variables of interest. Matched table codes are printed out, along with the Census-provided table description, the parsed variable names, and example table cells. The website `https://censusreporter.org/` may also be useful in finding variables.

Usage

```
find_dec_table(..., show = 2)

find_acs_table(..., show = 4)
```

Arguments

`...`	Variables to look for. These can be length-1 character vectors, or, for convenience, can be left unquoted (see examples).
`show`	How many matching tables to show. Increase this to show more possible matches, at the cost of more output. Negative values will be converted to positive but will suppress any printing.

Value

The codes for the top `show` tables, invisibly.

Examples

```
find_dec_table("sex", "age")
find_dec_table(tenure, race)
find_acs_table("income", "sex", show=3)
find_acs_table("heath care", show=-1)
```

get_table *Download data from a decennial census or ACS table*

Description

Leverages `tidycensus::get_decennial()` and `tidycensus::get_acs()` to download tables of census data. Tables are returned in tidy format, with variables given tidy, human-readable names.

Usage

```
get_dec_table(
  geography,
  table,
  state = NULL,
  county = NULL,
  ...,
  drop_total = FALSE
)

get_acs_table(
  geography,
  table,
  year = 2019,
  state = NULL,
  county = NULL,
  survey = c("acs5", "acs1"),
  ...,
  drop_total = FALSE
)
```

Arguments

geography
: The geography level to download data for. Usually one of `state`, `county`, `tract`, block group, `block`, `zcta`, etc. Consult https://walker-data.com/tidycensus/articles/basic-usage.html#geography-in-tidycensus for more information.

table
: The table code to download. See `find_dec_table()` for help identifying a table of interest. Note: some tables are split into A/B/C/etc. versions by race; this function unifies all of these tables under one code. So, for example, use `P012`, not `P012A`.

state
: The state to get data for, if any.

county
: The state to get data for, if any.

...
: Further arguments passed to `tidycensus::get_decennial()` or `tidycensus::get_acs()`, e.g. `year`, `state`, `county`, `geometry`.

drop_total
: Whether to filter out variables which are totals across another variable. Recommended only after inspection of the underlying table.

year
: For ACS data, the survey year to get data for.

survey
: For ACS data, whether to use the one-year or five-year survey (the default). Make sure to check availability using `find_acs_table()`.

Value

A tibble of census data in tidy format, with columns `GEOID`, `NAME`, `variable` (containing the Census variable code), `value` or `estiamte`, `moe` in the case of ACS tables, and additional factor columns specific to the table.

Examples

```
## Not run:
get_dec_table("state", "P003")
get_dec_table("state", "H002")
get_dec_table("county", "H002", state="WA", drop_total=TRUE)

get_acs_table("county subdivision", "B09001", state="WA", county="King")

## End(Not run)
```

marginalize *Helper function to sum over nuisance variables*

Description

For ACS data, margins of error will be updated appropriately, using tidycensus::moe_sum().

Usage

```
marginalize(data, ...)
```

Arguments

data	The output of get_dec_table() or get_acs_table()
...	The variables of interest, which will be kept. Remaining variables will be marginalized out.

Value

A new data frame that has had group_by() and summarize() applied.

Examples

```
## Not run:
d_cens = get_acs_table("state", "B25042")
marginalize(d_cens, bedrooms)

## End(Not run)
```

`tidiers` *Tidy labels in census tables*

Description

Some table labels are quite verbose, and users will often want to shorten them. These functions make tidying common types of labels easy. Most produce straightforward output, but there are several more generic tidiers:

- `tidy_simplify()` attempts to simplify labels by removing words common to all labels.
- `tidy_parens()` attempts to simplify labels by removing all terms in parentheses.

Usage

```
tidy_race(x)

tidy_ethnicity(x)

tidy_age(x)

tidy_age_bins(x, as_factor = FALSE)

tidy_simplify(x)

tidy_parens(x)
```

Arguments

`x`	A factor, which will be re-leveled. Character vectors will be converted to factors.
`as_factor`	if `TRUE`, return a factor with levels of the form [10,14].

Value

A re-leveled factor, except for `tidy_age_bins()`, which by default returns a data frame with columns `age_from` and `age_to` (inclusive).

Examples

```
ex_race_long = c("american indian and alaska native alone", "asian alone",
    "black or african american alone", "hispanic or latino",
    "native hawaiian and other pacific islander alone",
    "some other race alone", "total", "two or more races",
    "white alone", "white alone, not hispanic or latino")
tidy_race(ex_race_long)

tidy_age_bins(c("10 to 14 years", "21 years", "85 years and over"))

tidy_parens(c("label one (fake)", "label two (fake)"))
```

```
tidy_simplify(c("label one (fake)", "label two (fake)"))
```

Index