How to use `tidycensus`

Fernanda Sobrino

4/13/2021

Introduction

A simple exmaple

Make a map

Introduction

Introduction `tidycensus`

- why is this a useful package?
- installation and setup
- simple example
- finding variables
- get spatial data + make a map
- other features to explore
- you will need this package for Applied Pset 2

Motivation: get Census data directly into R

- the US Census Bureau publishes a ton of useful demographic data
- for example: population counts, poverty rates, household incomes, race/ethnic composition, educational attainment, commute mode . . .
- getting this data directly from the Census web page is annoying because:
  - data are not organized conveniently
  - every variable is in different tables
  - they are disaggregated by geographic level
  - you have to download a lot of `csv` files and spend too much time joining instead of analyzing

Census API + `tidycensus`: easier and nicer way

- API = Application Program Interface
- APIs have many purposes, but essencially they are like functions: user enters X and gets back Y
  - government and companies have servers that listen for requests and return the information
  - often requires an “API Key” to prevent overuse and to track users
- `tidycensus` is an R package that lets you write R code to query the API

What do I need to use `tidycensus`

- install the package: `isntall.packages("tidycensus")`
- get a census API key here
- the web page is (in case the link doesn't work) http://api.census.gov/data/key_signup.html
- use `census_api_key("YOUR KEY HERE", install = TRUE)` to permanently save it in your RStudio
- the use `Sys.getenv("CENSUS_API_KEY")` when you want to access it
- DON'T push the api key to github!!

A simple exmaple

Simple example

```r
library(tidycensus)
# load your API key
CENSUS_KEY <- Sys.getenv("CENSUS_API_KEY")
# get total population and income for all states from
# 5 year American Community Survey
acs_data <- get_acs(
  geography = "state",
  variables = c("B01001_001","B19013_001"),
  year = 2018
) %>%
  arrange(NAME)
```

```
## Getting data from the 2014-2018 5-year ACS
```

Simple example

```
head(acs_data)
```

```
## # A tibble: 6 x 5
##   GEOID NAME    variable   estimate   moe
##   <chr> <chr>   <chr>         <dbl> <dbl>
## 1 01    Alabama B01001_001  4864680    NA
## 2 01    Alabama B19013_001    48486   364
## 3 02    Alaska  B01001_001   738516    NA
## 4 02    Alaska  B19013_001    76715   894
## 5 04    Arizona B01001_001  6946685    NA
## 6 04    Arizona B19013_001    56213   275
```

Searching for variables

```r
# load a data frame of all possible variables and save it
# locally so it's fast next time
acs_vars_18 <- load_variables(2018, "acs5", cache = TRUE)
# you can use View() and search or use string
# matching to find what you need
head(acs_vars_18)
```

```
## # A tibble: 6 x 3
##   name     label             concept
##   <chr>    <chr>             <chr>
## 1 B00001_~ Estimate!!Total   UNWEIGHTED SAMPLE CO~
## 2 B00002_~ Estimate!!Total   UNWEIGHTED SAMPLE HO~
## 3 B01001_~ Estimate!!Total   SEX BY AGE
## 4 B01001_~ Estimate!!Total!~ SEX BY AGE
## 5 B01001_~ Estimate!!Total!~ SEX BY AGE
## 6 B01001_~ Estimate!!Total!~ SEX BY AGE
```

Request a whole table

```
# get counts of peoplle by education level by state
get_acs(
  geography = "state",
  table = "B15003",
) %>%
  arrange(NAME) %>%
  head()
```

```
## Getting data from the 2015-2019 5-year ACS

## Loading ACS5 variables for 2019 from table B15003. To ca

## # A tibble: 6 x 5
##   GEOID NAME    variable   estimate   moe
##   <chr> <chr>   <chr>         <dbl> <dbl>
## 1 01    Alabama B15003_001  3320877  1371
## 2 01    Alabama B15003_002    40837  1374
## 3 01    Alabama B15003_003      410   126
## 4 01    Alabama B15003_004      492   139
## 5 01    Alabama B15003_005      799   184
```

Make a map

Making maps with tidycensus

```
map <- get_acs(
  geography = "state",
  variables = "B19013_001",
  year = 2018,
  geometry = TRUE,
  shift_geo = TRUE
) %>%
  ggplot(aes(fill = estimate)) +
  geom_sf() +
  labs(title = "Median household income by state",
       caption = "Source: 2014-2018 ACS")
```

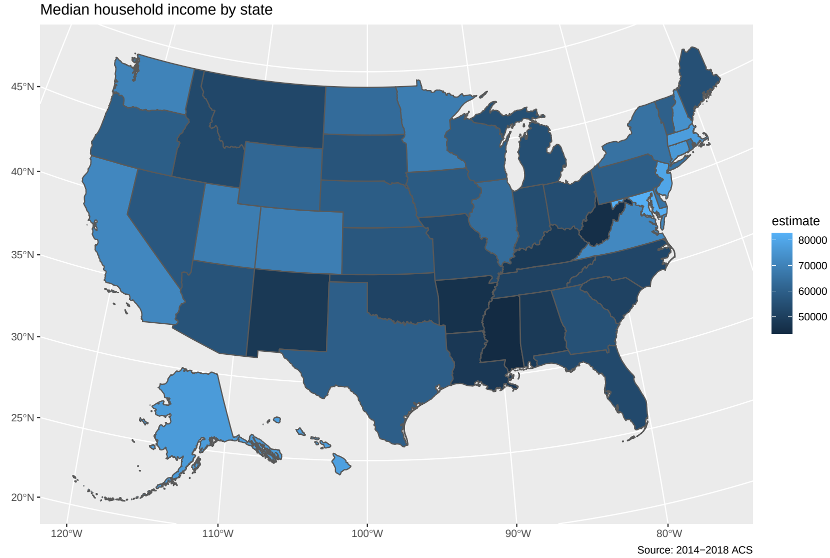
```
## Getting data from the 2014-2018 5-year ACS

## Using feature geometry obtained from the albersusa packa

## Please note: Alaska and Hawaii are being shifted and are
```

Making maps with `tidycensus`

```
map
```



More tools

- see here: https://walker-data.com/tidycensus/index.html
- other geographies: countries, tracts, ZIPs, many more
- `state` and `country` argument in `get_acs` filters to specific state
    - example: Census tracts in Cook Country, IL
- `get_decennial()` for data from 10-year Censuses
- `fips_codes` data frame for matching geography codes and names
- request different confidence intervals and compute statistical significance