Introduction to Census Data in R

NCEAS Learning Hub

for

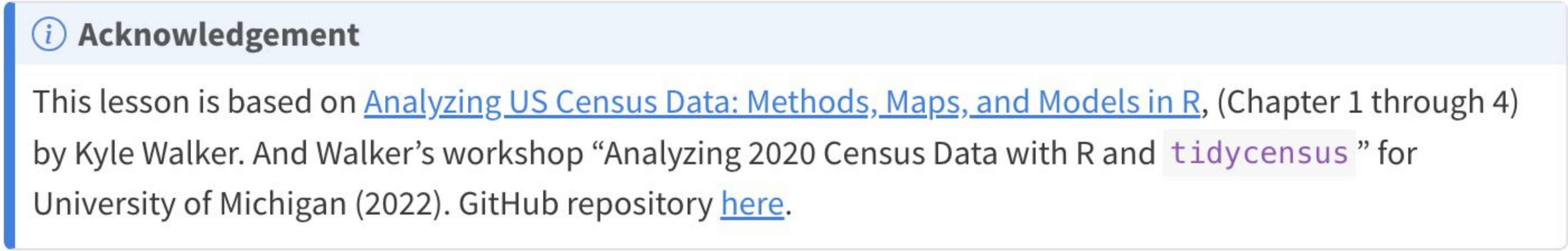
Fundamentals in Data Management for Qualitative and Quantitative Arctic Research

January 2024

Learning Objectives

- Provide an overview of US Census data
- Introduce the main functions of the `tidycensus` package to be able to work with census data
- Review data wrangling function to get census data ready for analysis
- Plot census data using `ggplot2`

This lesson is based on..

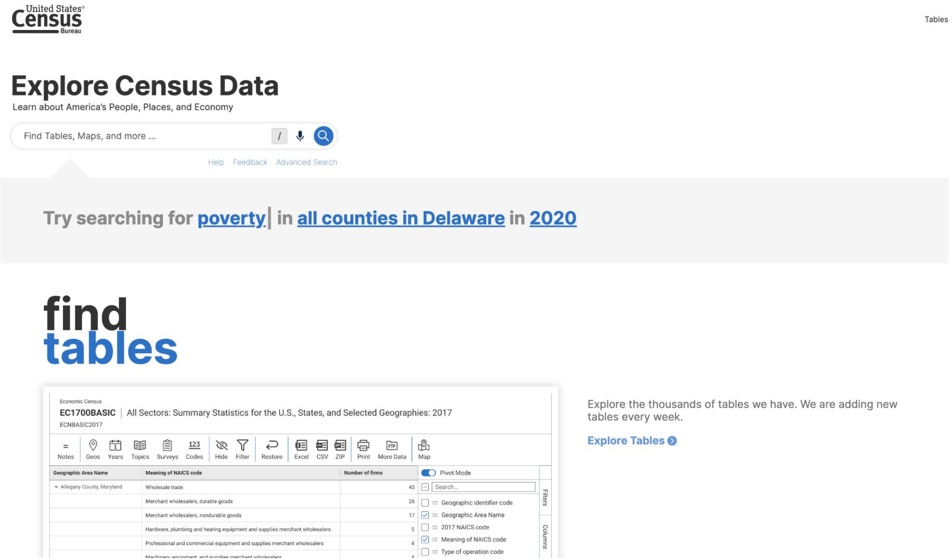


Acknowledgement

This lesson is based on Analyzing US Census Data: Methods, Maps, and Models in R, (Chapter 1 through 4) by Kyle Walker. And Walker's workshop "Analyzing 2020 Census Data with R and `tidycensus`" for University of Michigan (2022). GitHub repository here.

General Workflow

data.census.gov



Tables

Explore Census Data

Learn about America's People, Places, and Economy

Find Tables, Maps, and more ...

Help Feedback Advanced Search

Try searching for poverty in all counties in Delaware in 2020

find tables

Economic Census

EC1700BASIC | All Sectors: Summary Statistics for the U.S., States, and Selected Geographies: 2017

ECNBASIC2017

Geographic Area Name	Meaning of NAICS code	Number of firms
Allegany County, Maryland	Wholesale trade	43
	Merchant wholesalers, durable goods	26
	Merchant wholesalers, nondurable goods	17
	Hardware, plumbing and heating equipment and supplies merchant wholesalers	5
	Professional and commercial equipment and supplies merchant wholesalers	4
	Machinery, equipment, and supplies merchant wholesalers	4

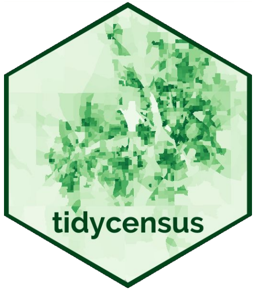
Pivot Mode

□ Geographic identifier code
☑ Geographic Area Name
□ 2017 NAICS code
☑ Meaning of NAICS code
□ Type of operation code

Explore the thousands of tables we have. We are adding new tables every week.

Explore Tables

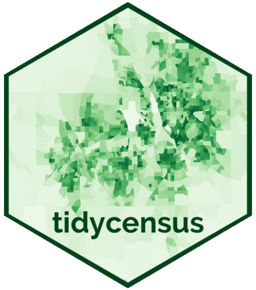
But... `tidycensus`



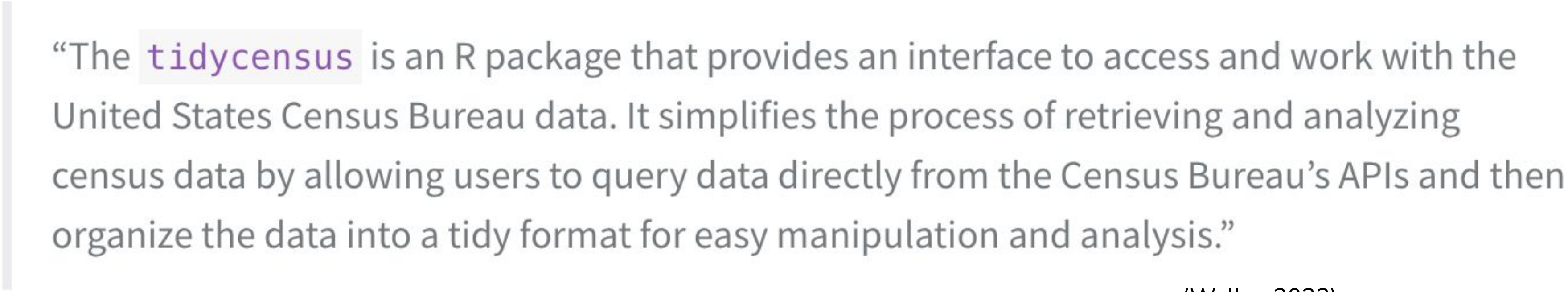
(Walker and Herman 2021)

- Was developed to systematize this process and do this systematization using R.
- Idea came to be after constantly using same function over and over
- Census Application Programming Interface (API)
- The vision behind this package was to incorporate the API access into an R package to facilitate access to census data using R

But... `tidycensus`



(Walker and Herman 2021)



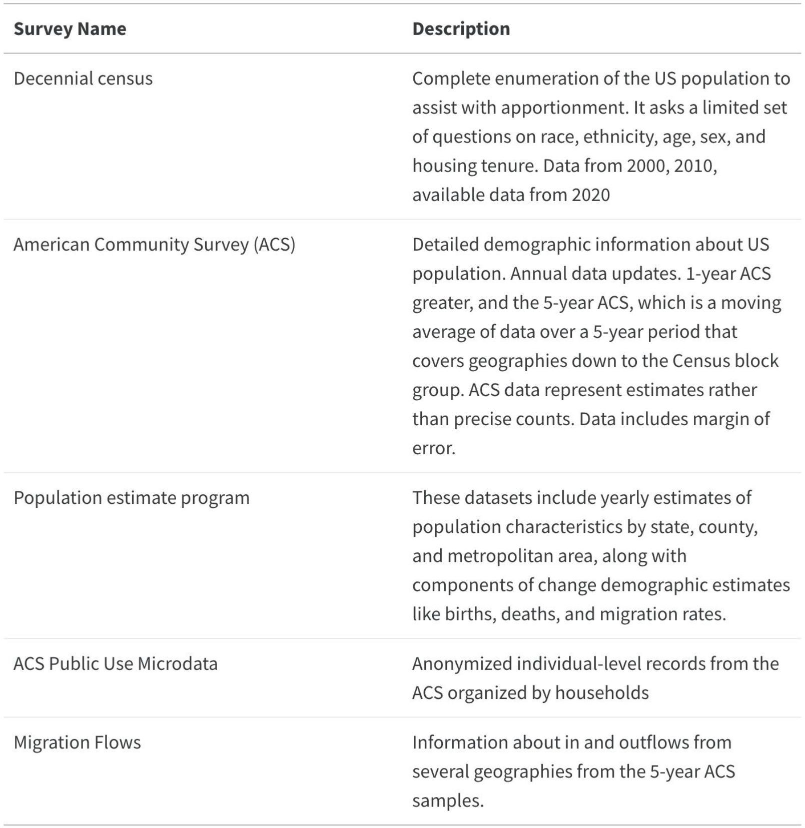
> “The `tidycensus` is an R package that provides an interface to access and work with the United States Census Bureau data. It simplifies the process of retrieving and analyzing census data by allowing users to query data directly from the Census Bureau’s APIs and then organize the data into a tidy format for easy manipulation and analysis.”

(Walker 2023)

General Structure

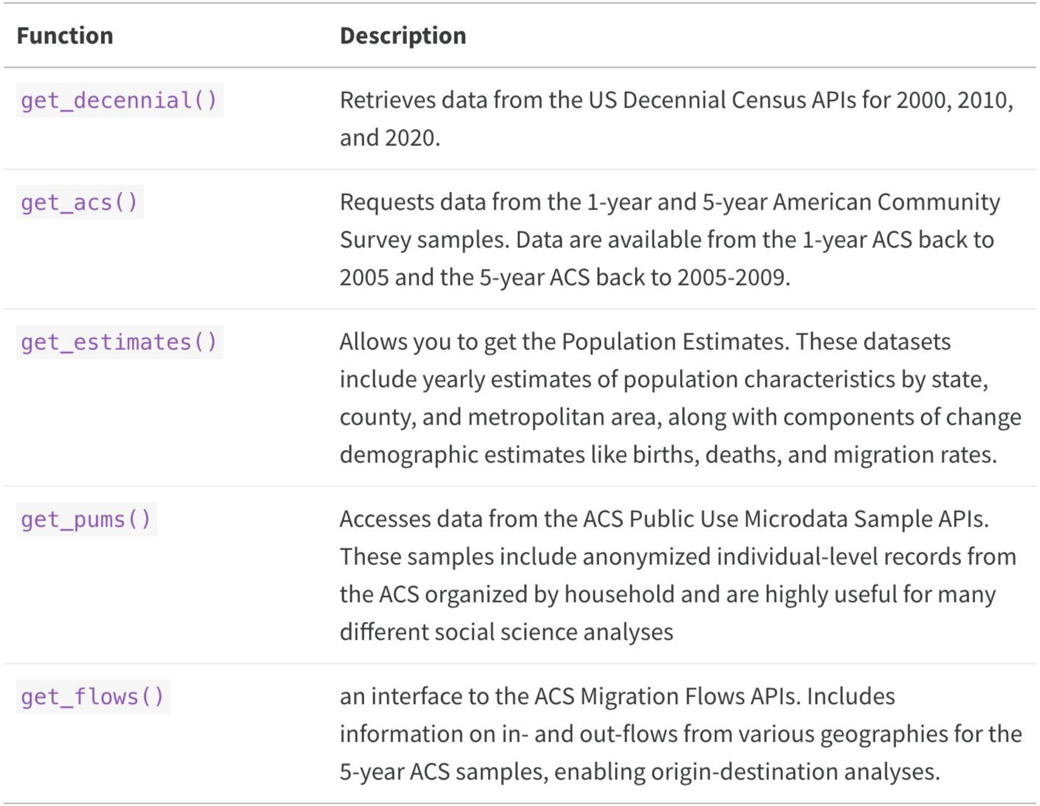
- `tidycensus` takes an opinionated approach to accessing a selected number of census APIs. The main goal is to facilitate access to a few census APIs through R
- The idea behind this package is to make the tedious process of working with Census data more concise. It pulls data from the census API and returns it to the user in a "tidy" format
- Can easily merge census geometries to data for mapping. Which apparently can be a very time-consuming task
- Includes tools for handling margins of errors in the ACS and working with survey weights in the ACS Public Use Microdata
- You can request data from states and counties by name instead of FIPS codes.

Information you can access



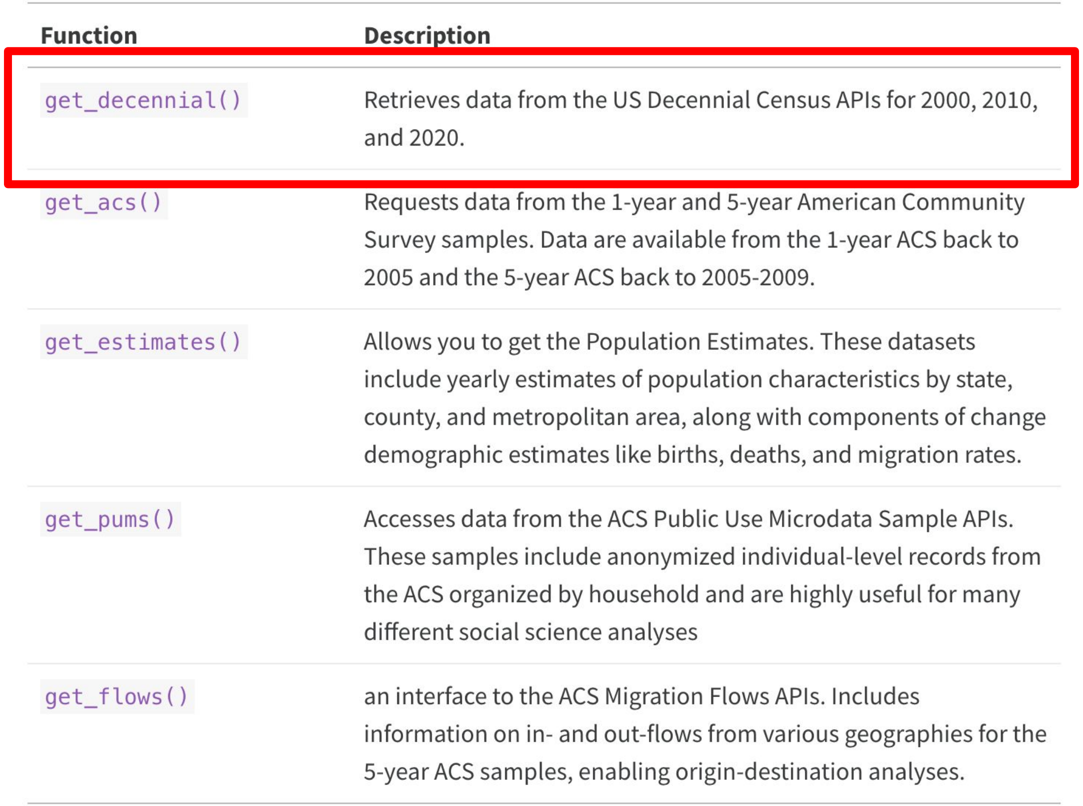
Survey Name	Description
Decennial census	Complete enumeration of the US population to assist with apportionment. It asks a limited set of questions on race, ethnicity, age, sex, and housing tenure. Data from 2000, 2010, available data from 2020
American Community Survey (ACS)	Detailed demographic information about US population. Annual data updates. 1-year ACS greater, and the 5-year ACS, which is a moving average of data over a 5-year period that covers geographies down to the Census block group. ACS data represent estimates rather than precise counts. Data includes margin of error.
Population estimate program	These datasets include yearly estimates of population characteristics by state, county, and metropolitan area, along with components of change demographic estimates like births, deaths, and migration rates.
ACS Public Use Microdata	Anonymized individual-level records from the ACS organized by households
Migration Flows	Information about in and outflows from several geographies from the 5-year ACS samples.

Getting Census Data (Core functions)



Function	Description
`get_decennial()`	Retrieves data from the US Decennial Census APIs for 2000, 2010, and 2020.
`get_acs()`	Requests data from the 1-year and 5-year American Community Survey samples. Data are available from the 1-year ACS back to 2005 and the 5-year ACS back to 2005-2009.
`get_estimates()`	Allows you to get the Population Estimates. These datasets include yearly estimates of population characteristics by state, county, and metropolitan area, along with components of change demographic estimates like births, deaths, and migration rates.
`get_pums()`	Accesses data from the ACS Public Use Microdata Sample APIs. These samples include anonymized individual-level records from the ACS organized by household and are highly useful for many different social science analyses
`get_flows()`	an interface to the ACS Migration Flows APIs. Includes information on in- and out-flows from various geographies for the 5-year ACS samples, enabling origin-destination analyses.

Getting Census Data (Core functions)

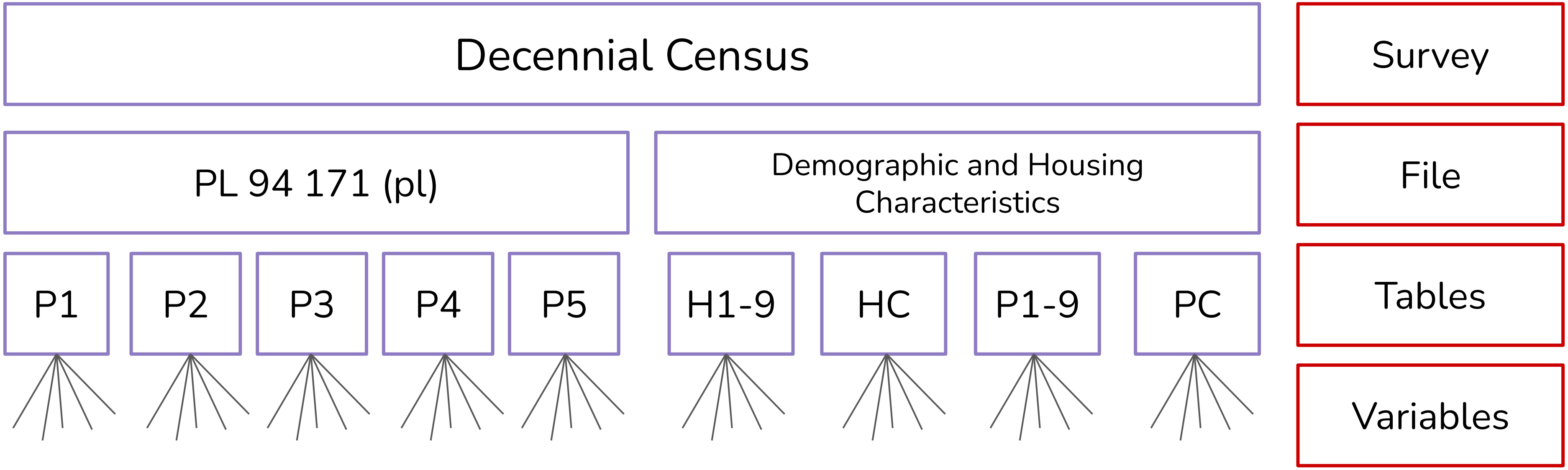


Function	Description
`get_decennial()`	Retrieves data from the US Decennial Census APIs for 2000, 2010, and 2020.
`get_acs()`	Requests data from the 1-year and 5-year American Community Survey samples. Data are available from the 1-year ACS back to 2005 and the 5-year ACS back to 2005-2009.
`get_estimates()`	Allows you to get the Population Estimates. These datasets include yearly estimates of population characteristics by state, county, and metropolitan area, along with components of change demographic estimates like births, deaths, and migration rates.
`get_pums()`	Accesses data from the ACS Public Use Microdata Sample APIs. These samples include anonymized individual-level records from the ACS organized by household and are highly useful for many different social science analyses
`get_flows()`	an interface to the ACS Migration Flows APIs. Includes information on in- and out-flows from various geographies for the 5-year ACS samples, enabling origin-destination analyses.

2020 Decennial Survey

- Challenges of running a census during a pandemic
- Data release has been delayed
- One of the main files from the 2020 census is the **PL 94-171** Redistricting Summary File which is used for congressional appointments and redistricting. Variable available in this file are:
  - Total counts (population & households)
  - Occupied/vacant housing unit
  - Total and voting age population breakdown by race & ethnicity
- Another important file is the **Demographic and Housing Characteristics** Summary Files (Different to summary file 1 form 2010). Contains age and sex breakdowns and detailed race and ethnicity data.

Census Data Organization



Decennial Census

PL 94 171 (pl)

Demographic and Housing Characteristics

P1

P2

P3

P4

P5

H1-9

HC

P1-9

PC

Survey

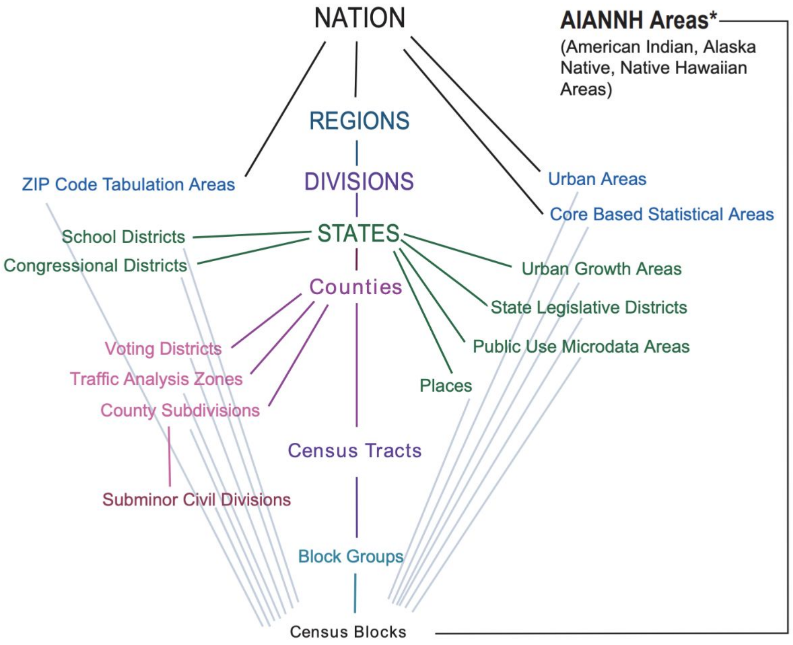
File

Tables

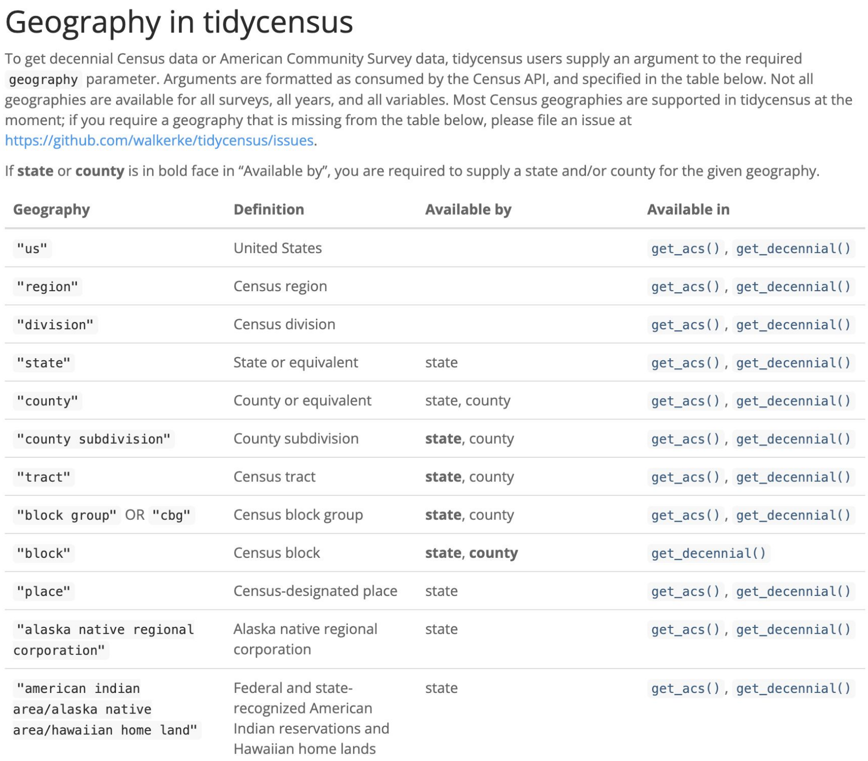
Variables

Census Data Organization

Census Geography: types of geographic areas used by the Census Bureau in its data collection and tabulation operations



More on Census Geographies



Geography in tidycensus

To get decennial Census data or American Community Survey data, tidycensus users supply an argument to the required `geography` parameter. Arguments are formatted as consumed by the Census API, and specified in the table below. Not all geographies are available for all surveys, all years, and all variables. Most Census geographies are supported in tidycensus at the moment; if you require a geography that is missing from the table below, please file an issue at https://github.com/walkerke/tidycensus/issues.

If **state** or **county** is in bold face in "Available by", you are required to supply a state and/or county for the given geography.

Geography	Definition	Available by	Available in
`"us"`	United States		`get_acs()`, `get_decennial()`
`"region"`	Census region		`get_acs()`, `get_decennial()`
`"division"`	Census division		`get_acs()`, `get_decennial()`
`"state"`	State or equivalent	state	`get_acs()`, `get_decennial()`
`"county"`	County or equivalent	state, county	`get_acs()`, `get_decennial()`
`"county subdivision"`	County subdivision	**state**, county	`get_acs()`, `get_decennial()`
`"tract"`	Census tract	**state**, county	`get_acs()`, `get_decennial()`
`"block group"` OR `"cbg"`	Census block group	**state**, county	`get_acs()`, `get_decennial()`
`"block"`	Census block	**state**, **county**	`get_decennial()`
`"place"`	Census-designated place	state	`get_acs()`, `get_decennial()`
`"alaska native regional corporation"`	Alaska native regional corporation	state	`get_acs()`, `get_decennial()`
`"american indian area/alaska native area/hawaiian home land"`	Federal and state-recognized American Indian reservations and Hawaiian home lands	state	`get_acs()`, `get_decennial()`

https://walker-data.com/tidycensus/articles/basic-usage.html#geography-in-tidycensus

get_decennial()

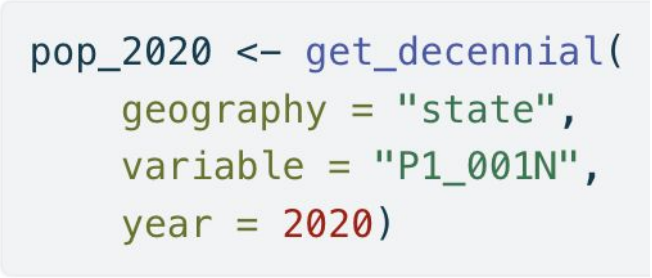
`get_decennial()`

- Geography
- Variable
- Year
- Sumfile

get_decennial()

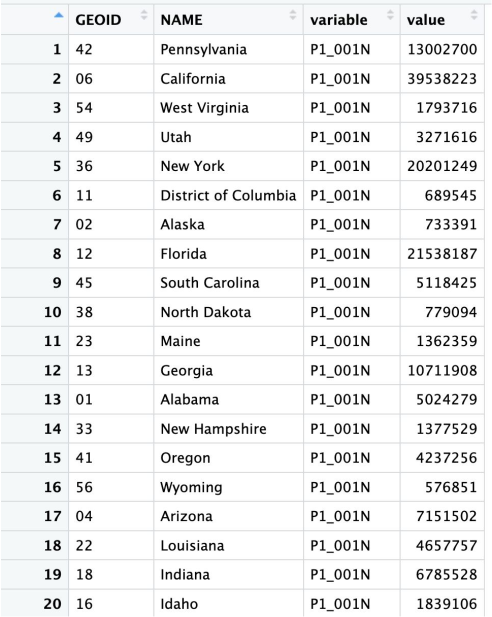
get_decennial()

- Geography
- Variable
- Year
- Sumfile



```
pop_2020 <- get_decennial(
    geography = "state",
    variable = "P1_001N",
    year = 2020)
```

Default = “pl”

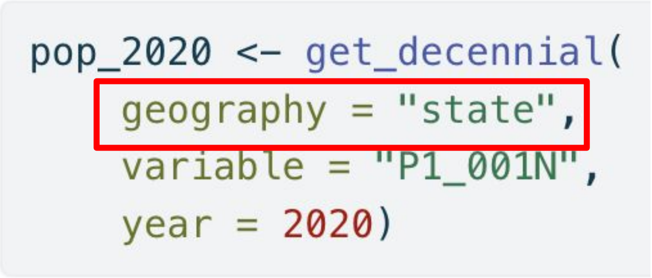


	GEOID	NAME	variable	value
1	42	Pennsylvania	P1_001N	13002700
2	06	California	P1_001N	39538223
3	54	West Virginia	P1_001N	1793716
4	49	Utah	P1_001N	3271616
5	36	New York	P1_001N	20201249
6	11	District of Columbia	P1_001N	689545
7	02	Alaska	P1_001N	733391
8	12	Florida	P1_001N	21538187
9	45	South Carolina	P1_001N	5118425
10	38	North Dakota	P1_001N	779094
11	23	Maine	P1_001N	1362359
12	13	Georgia	P1_001N	10711908
13	01	Alabama	P1_001N	5024279
14	33	New Hampshire	P1_001N	1377529
15	41	Oregon	P1_001N	4237256
16	56	Wyoming	P1_001N	576851
17	04	Arizona	P1_001N	7151502
18	22	Louisiana	P1_001N	4657757
19	18	Indiana	P1_001N	6785528
20	16	Idaho	P1_001N	1839106

get_decennial()

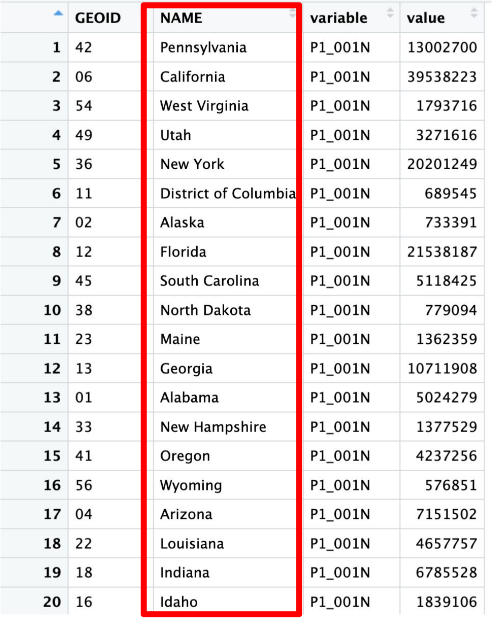
`get_decennial()`

- ○ **Geography**
- ○ Variable
- ○ Year
- ○ Sumfile



```
pop_2020 <- get_decennial(
    geography = "state",
    variable = "P1_001N",
    year = 2020)
```

`Default = “pl”`



	GEOID	NAME	variable	value
1	42	Pennsylvania	P1_001N	13002700
2	06	California	P1_001N	39538223
3	54	West Virginia	P1_001N	1793716
4	49	Utah	P1_001N	3271616
5	36	New York	P1_001N	20201249
6	11	District of Columbia	P1_001N	689545
7	02	Alaska	P1_001N	733391
8	12	Florida	P1_001N	21538187
9	45	South Carolina	P1_001N	5118425
10	38	North Dakota	P1_001N	779094
11	23	Maine	P1_001N	1362359
12	13	Georgia	P1_001N	10711908
13	01	Alabama	P1_001N	5024279
14	33	New Hampshire	P1_001N	1377529
15	41	Oregon	P1_001N	4237256
16	56	Wyoming	P1_001N	576851
17	04	Arizona	P1_001N	7151502
18	22	Louisiana	P1_001N	4657757
19	18	Indiana	P1_001N	6785528
20	16	Idaho	P1_001N	1839106

Census Geography: types of geographic areas used by the Census Bureau in its data collection and tabulation operations

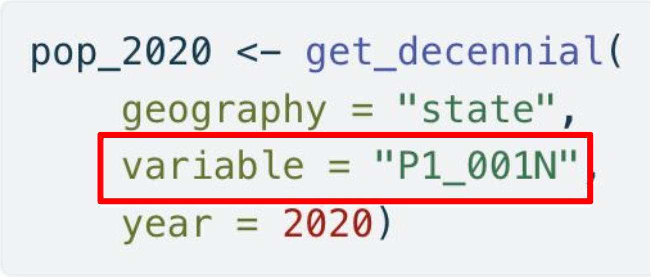
get_decennial()

get_decennial()

- ○ Geography
- ○ **Variable**
- ○ Year
- ○ Sumfile

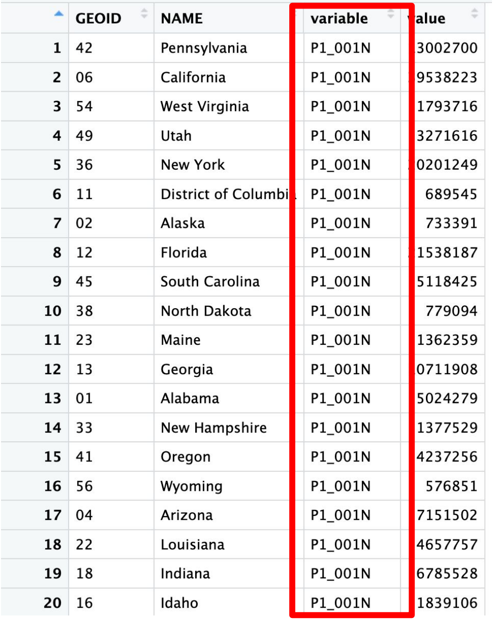
E.g.

- P1_001N = Total Population
- P2_002N = Total population Hispanic or Latino



```
pop_2020 <- get_decennial(
    geography = "state",
    variable = "P1_001N",
    year = 2020)
```

Default = “pl”

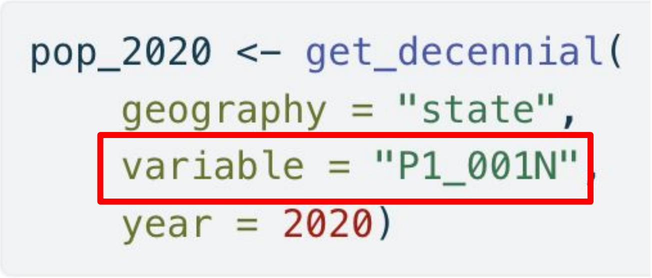


	GEOID	NAME	variable	value
1	42	Pennsylvania	P1_001N	3002700
2	06	California	P1_001N	9538223
3	54	West Virginia	P1_001N	1793716
4	49	Utah	P1_001N	3271616
5	36	New York	P1_001N	0201249
6	11	District of Columbi	P1_001N	689545
7	02	Alaska	P1_001N	733391
8	12	Florida	P1_001N	1538187
9	45	South Carolina	P1_001N	5118425
10	38	North Dakota	P1_001N	779094
11	23	Maine	P1_001N	1362359
12	13	Georgia	P1_001N	0711908
13	01	Alabama	P1_001N	5024279
14	33	New Hampshire	P1_001N	1377529
15	41	Oregon	P1_001N	4237256
16	56	Wyoming	P1_001N	576851
17	04	Arizona	P1_001N	7151502
18	22	Louisiana	P1_001N	4657757
19	18	Indiana	P1_001N	6785528
20	16	Idaho	P1_001N	1839106

get_decennial()

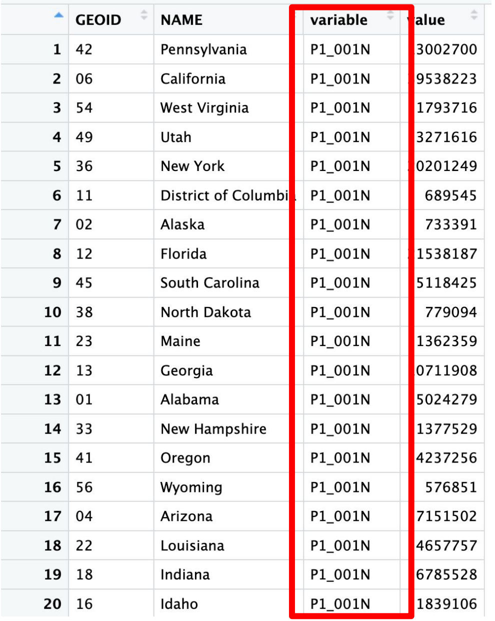
get_decennial()

- ○ Geography
- ○ Variable
- ○ Year
- ○ Sumfile



```
pop_2020 <- get_decennial(
    geography = "state",
    variable = "P1_001N",
    year = 2020)
```

Default = “pl”



	GEOID	NAME	variable	alue
1	42	Pennsylvania	P1_001N	3002700
2	06	California	P1_001N	9538223
3	54	West Virginia	P1_001N	1793716
4	49	Utah	P1_001N	3271616
5	36	New York	P1_001N	0201249
6	11	District of Columbi	P1_001N	689545
7	02	Alaska	P1_001N	733391
8	12	Florida	P1_001N	1538187
9	45	South Carolina	P1_001N	5118425
10	38	North Dakota	P1_001N	779094
11	23	Maine	P1_001N	1362359
12	13	Georgia	P1_001N	0711908
13	01	Alabama	P1_001N	5024279
14	33	New Hampshire	P1_001N	1377529
15	41	Oregon	P1_001N	4237256
16	56	Wyoming	P1_001N	576851
17	04	Arizona	P1_001N	7151502
18	22	Louisiana	P1_001N	4657757
19	18	Indiana	P1_001N	6785528
20	16	Idaho	P1_001N	1839106

E.g.

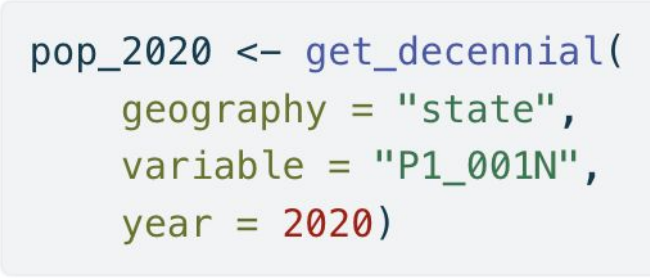
- P1_001N = Total Population
- P2_002N = Total population Hispanic or Latino

- You can query more than one variable with `variable =`
  - ○ `variable = c("P2_002N", "P2_006N")`

get_decennial()

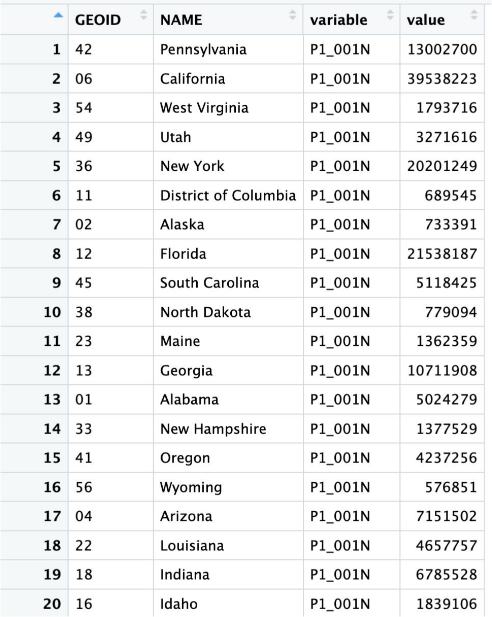
get_decennial()

- ○ Geography
- ○ Variable
- ○ Year
- ○ **Sumfile**



```
pop_2020 <- get_decennial(
    geography = "state",
    variable = "P1_001N",
    year = 2020)
```

Default = “pl”
OR specify
● “dhc”

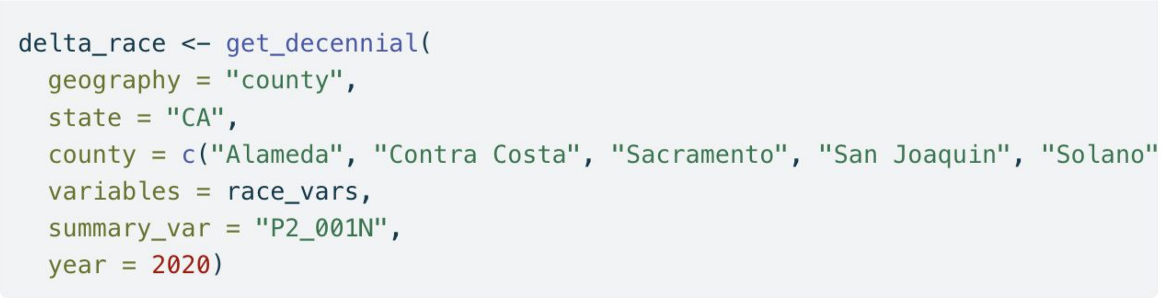


	GEOID	NAME	variable	value
1	42	Pennsylvania	P1_001N	13002700
2	06	California	P1_001N	39538223
3	54	West Virginia	P1_001N	1793716
4	49	Utah	P1_001N	3271616
5	36	New York	P1_001N	20201249
6	11	District of Columbia	P1_001N	689545
7	02	Alaska	P1_001N	733391
8	12	Florida	P1_001N	21538187
9	45	South Carolina	P1_001N	5118425
10	38	North Dakota	P1_001N	779094
11	23	Maine	P1_001N	1362359
12	13	Georgia	P1_001N	10711908
13	01	Alabama	P1_001N	5024279
14	33	New Hampshire	P1_001N	1377529
15	41	Oregon	P1_001N	4237256
16	56	Wyoming	P1_001N	576851
17	04	Arizona	P1_001N	7151502
18	22	Louisiana	P1_001N	4657757
19	18	Indiana	P1_001N	6785528
20	16	Idaho	P1_001N	1839106

WHICH FILE

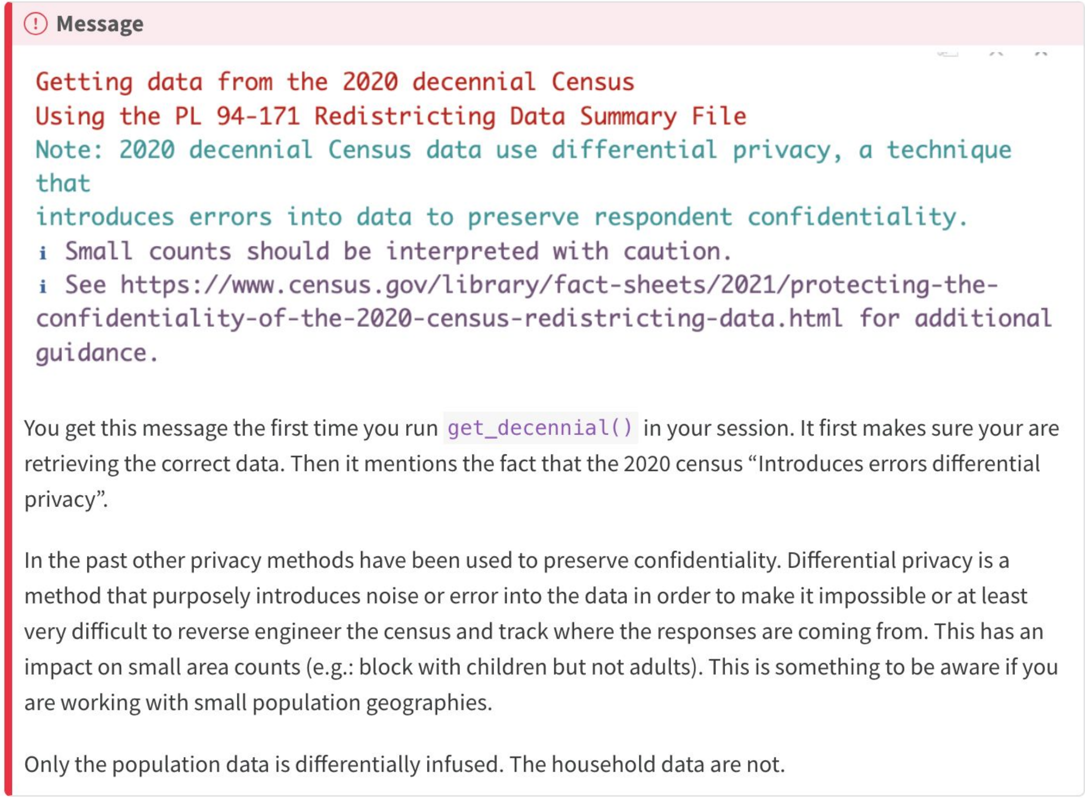
Filtering Arguments

- You can filter by one or more states with `state =`
  - `state = “CA”`
  - `state = c(“CA”, “WA”)`
- You can filter by one or more counties with `county =`
  - `county = "Santa Barbara"`
  - `county = c("Santa Barbara", “Ventura” “San Luis Obispos”)`
- Everytime you specify more than one *thing*, you use the `c()` function



```
delta_race <- get_decennial(
  geography = "county",
  state = "CA",
  county = c("Alameda", "Contra Costa", "Sacramento", "San Joaquin", "Solano"
  variables = race_vars,
  summary_var = "P2_001N",
  year = 2020)
```

get_decennial()



Message

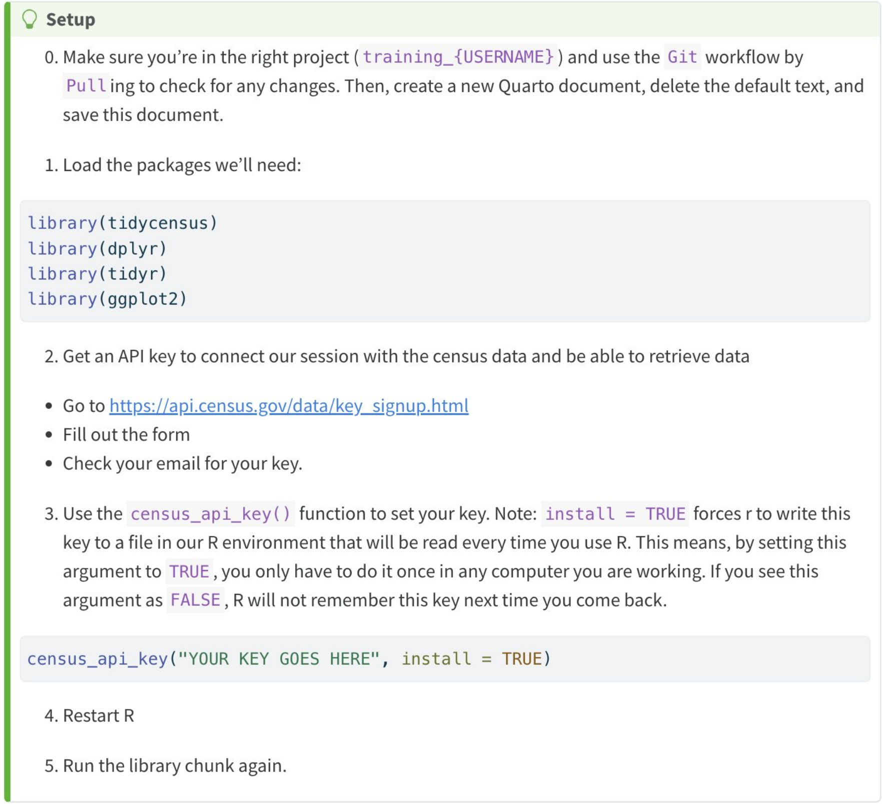
```
Getting data from the 2020 decennial Census
Using the PL 94-171 Redistricting Data Summary File
Note: 2020 decennial Census data use differential privacy, a technique
that
introduces errors into data to preserve respondent confidentiality.
ℹ Small counts should be interpreted with caution.
ℹ See https://www.census.gov/library/fact-sheets/2021/protecting-the-
confidentiality-of-the-2020-census-redistricting-data.html for additional
guidance.
```

You get this message the first time you run `get_decennial()` in your session. It first makes sure your are retrieving the correct data. Then it mentions the fact that the 2020 census “Introduces errors differential privacy”.

In the past other privacy methods have been used to preserve confidentiality. Differential privacy is a method that purposely introduces noise or error into the data in order to make it impossible or at least very difficult to reverse engineer the census and track where the responses are coming from. This has an impact on small area counts (e.g.: block with children but not adults). This is something to be aware if you are working with small population geographies.

Only the population data is differentially infused. The household data are not.

Getting Started



Setup

0. Make sure you're in the right project (`training_{USERNAME}`) and use the `Git` workflow by `Pull`ing to check for any changes. Then, create a new Quarto document, delete the default text, and save this document.

1. Load the packages we'll need:

```r
library(tidycensus)
library(dplyr)
library(tidyr)
library(ggplot2)
```

2. Get an API key to connect our session with the census data and be able to retrieve data

- Go to https://api.census.gov/data/key_signup.html
- Fill out the form
- Check your email for your key.

3. Use the `census_api_key()` function to set your key. Note: `install = TRUE` forces r to write this key to a file in our R environment that will be read every time you use R. This means, by setting this argument to `TRUE`, you only have to do it once in any computer you are working. If you see this argument as `FALSE`, R will not remember this key next time you come back.

```r
census_api_key("YOUR KEY GOES HERE", install = TRUE)
```

4. Restart R

5. Run the library chunk again.

GET API KEY
Go to:
https://api.census.gov/data/key_signup.html

American Community Survey (ACS)

- Since 2010, it replaced the Census *long form* (details about demographics, income, education, language, etc).

(Walker 2023)

American Community Survey (ACS)

- Since 2010, it replaced the Census *long form* (details about demographics, income, education, language, etc)
- Primary source of detailed demographic information about the US population

(Walker 2023)

American Community Survey (ACS)

- Since 2010, it replaced the Census *long form* (details about demographics, income, education, language, etc)
- Primary source of detailed demographic information about the US population
- Is mailed to approximately 3.5 million households **per year** (representing around 3 percent of the US population)

(Walker 2023)

American Community Survey (ACS)

- Since 2010, it replaced the Census *long form* (details about demographics, income, education, language, etc)
- Primary source of detailed demographic information about the US population
- Is mailed to approximately 3.5 million households **per year** (representing around 3% of the US population)
- The Census Bureau releases two ACS datasets to the public: the **1-year ACS**, which covers areas of population 65,000 and greater,

(Walker 2023)

American Community Survey (ACS)

- Since 2010, it replaced the Census *long form* (details about demographics, income, education, language, etc)
- Primary source of detailed demographic information about the US population
- Is mailed to approximately 3.5 million households **per year** (representing around 3 percent of the US population)
- The Census Bureau releases two ACS datasets to the public: the **1-year ACS**, which covers areas of population 65,000 and greater,
- And the **5-year** ACS, which is a moving average of data over a 5-year period that covers geographies down to the Census block group

(Walker 2023)

American Community Survey (ACS)

- Since 2010, it replaced the Census *long form* (details about demographics, income, education, language, etc)
- Primary source of detailed demographic information about the US population
- Is mailed to approximately 3.5 million households **per year** (representing around 3 percent of the US population)
- The Census Bureau releases two ACS datasets to the public: the **1-year ACS**, which covers areas of population 65,000 and greater,
- And the **5-year** ACS, which is a moving average of data over a 5-year period that covers geographies down to the Census block group
- ACS data are distinct from decennial Census data in that data represent *estimates* rather than precise counts, and in turn are characterized by ***margins of error*** around those estimates.

(Walker 2023)

American Community Survey (ACS)

- Since 2010, it replaced the Census *long form* (details about demographics, income, education, language, etc)
- Primary source of detailed demographic information about the US population
- Is mailed to approximately 3.5 million households **per year** (representing around 3 percent of the US population)
- The Census Bureau releases two ACS datasets to the public: the **1-year ACS**, which covers areas of population 65,000 and greater,
- And the **5-year** ACS, which is a moving average of data over a 5-year period that covers geographies down to the Census block group
- ACS data are distinct from decennial Census data in that data represent *estimates* rather than precise counts, and in turn are characterized by *margins of error* around those estimates.
- 2020 1-year ACS data will not be released, instead, replaced by experimental estimates for that year. Because of data collection issues during the COVID-19 pandemic.

(Walker 2023)

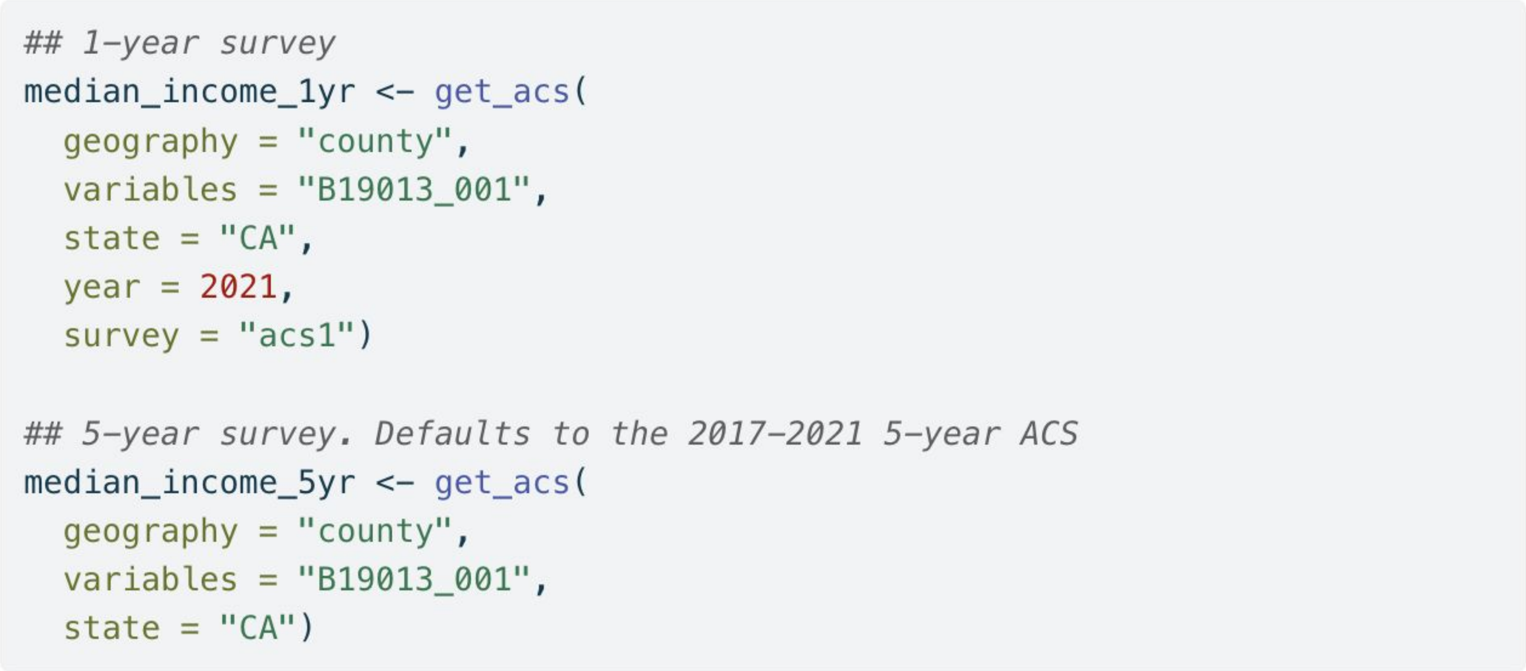
`get_acs()`

- The functions operates very similar to `get_decennial()`
- The main differences is that it access a different survey so **the options for each argument change.**
- The two required arguments are `geography` and `variables`. The function defaults to the **2017-2021 5-year ACS**
- 1-year ACS data are more current, but are only available for geographies of population 65,000 and greater
- Access 1-year ACS data with the argument `survey = "acs1"`; defaults to “acs5”

`?get_acs()`

- Three main arguments
  - Geography
    - The geographic area of your data
  - Variable(s)
    - Character string or vector of character strings of variable IDs. `tidycensus` automatically returns the estimate and the margin of error associated with the variable.
  - Year
    - The year, or endyear, of the ACS sample. 5-year ACS data is available from 2009 through 2021; 1-year ACS data is available from 2005 through 2021, with the exception of 2020. Defaults to 2021.

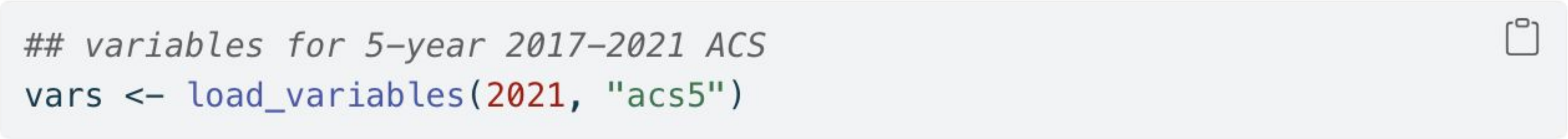
get_acs()



```r
## 1-year survey
median_income_1yr <- get_acs(
  geography = "county",
  variables = "B19013_001",
  state = "CA",
  year = 2021,
  survey = "acs1")

## 5-year survey. Defaults to the 2017-2021 5-year ACS
median_income_5yr <- get_acs(
  geography = "county",
  variables = "B19013_001",
  state = "CA")
```

ACS `load_variables()`



```r
## variables for 5-year 2017-2021 ACS
vars <- load_variables(2021, "acs5")
```

Practice