Reproducible Data Retrieval: Pulling Data Directly into R

In the last few modules, we have cleaned data that were pulled from various sources and you were directed to calculate specific metrics. In this module, you will learn how to pull data for yourself and use that data to calculate metrics that you hypothesize may be important.

Objectives:

1. Navigate the US Census API to locate variables of interest from the American Community Survey.
2. Use the `tidycensus` package to pull data from the American Community Survey and calculate a descriptive metric of interest.
3. Use the NASS Quick Stats page to locate variables of interest from the Census of Agriculture.
4. Use the `rnassqs` package to pull data from the Census of Agriculture and calculate a descriptive metric of interest.

Getting Started

(1) Open RStudio and your project file. (2) Create a new Quarto file.

American Community Survey

In the last module, we used the `tidycensus` package to pull data on the number of retirees based on the code assigned to each piece of information. If you want to pull your own data, you will first need to find the correct codes.

All the codes for the American Community Survey can be found online at https://api.census.gov/data/2017/acs/acs5/groups.html. Note that you can substitute 2017 in this URL with any year from 2009-2023. Since we have been working with 2017 data from the Census of Agriculture, we use 2017 here for consistency.

When you go to this webpage, you will see a table with a table code, a table description, and a link to a variable list. First, choose a description that you'd like to take a closer look at. For this example, I'll look at a table with information on citizenship. The description is NATIVITY AND CITIZENSHIP IN THE UNITED STATES and the table code is B05001.

If I want the whole table, I can use `tidycensus` to pull in the whole table.

(3) Import `tidycensus` library. (4) Import the Census API key you received in Module 5 with `census_api_key("YOUR API KEY GOES HERE")`. (5) Hide your API key in the report with `{r, include=FALSE}`.

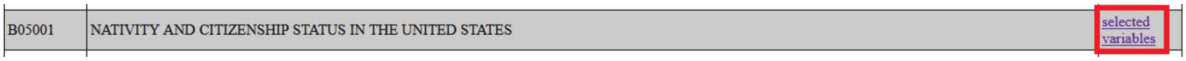
(6) Pull in the nativity and citizenship table with the code below:

```
citizens <- get_acs(geography = "county",
                    state = "ID",
                    county = "Bear Lake", # your county here
                    year = 2017,
                    survey = "acs5",
                    table = "B05001")
```

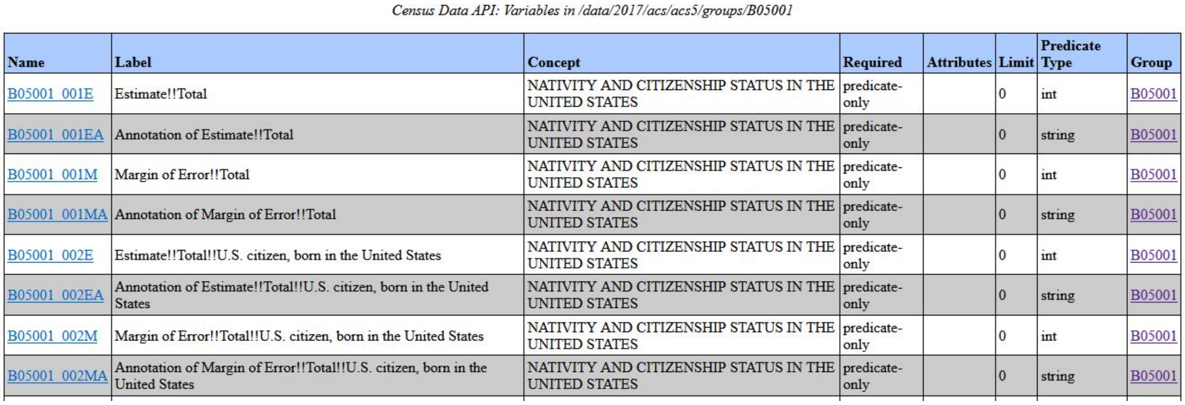
(7) Print the `citizens` data.

(8) You should see 5 columns: GEOID, NAME, variable, estimate, moe. GEOID stands for geographic identifier, a unique code that applies to your county. NAME is the name of the GEO ID – the name of your county, in this case. variable is the code indicating the information contained in each row. estimate is the number of people in each category. moe stands for margin of error and estimates how accurate the estimate is. We will not use moe in this analysis.

Now that you have a table, you need to know what those variables are! (9) Click the "selected variables" link for the B05001 table on the American Community Survey API website to see the variable codes.



B05001	NATIVITY AND CITIZENSHIP STATUS IN THE UNITED STATES	selected variables



Census Data API: Variables in /data/2017/acs/acs5/groups/B05001

Name	Label	Concept	Required	Attributes	Limit	Predicate Type	Group
B05001_001E	Estimate!!Total	NATIVITY AND CITIZENSHIP STATUS IN THE UNITED STATES	predicate-only		0	int	B05001
B05001_001EA	Annotation of Estimate!!Total	NATIVITY AND CITIZENSHIP STATUS IN THE UNITED STATES	predicate-only		0	string	B05001
B05001_001M	Margin of Error!!Total	NATIVITY AND CITIZENSHIP STATUS IN THE UNITED STATES	predicate-only		0	int	B05001
B05001_001MA	Annotation of Margin of Error!!Total	NATIVITY AND CITIZENSHIP STATUS IN THE UNITED STATES	predicate-only		0	string	B05001
B05001_002E	Estimate!!Total!!U.S. citizen, born in the United States	NATIVITY AND CITIZENSHIP STATUS IN THE UNITED STATES	predicate-only		0	int	B05001
B05001_002EA	Annotation of Estimate!!Total!!U.S. citizen, born in the United States	NATIVITY AND CITIZENSHIP STATUS IN THE UNITED STATES	predicate-only		0	string	B05001
B05001_002M	Margin of Error!!Total!!U.S. citizen, born in the United States	NATIVITY AND CITIZENSHIP STATUS IN THE UNITED STATES	predicate-only		0	int	B05001
B05001_002MA	Annotation of Margin of Error!!Total!!U.S. citizen, born in the United States	NATIVITY AND CITIZENSHIP STATUS IN THE UNITED STATES	predicate-only		0	string	B05001

Each code appears in this table four times. For example, variable B05001_001 appears as B05001_001E, B05001_001EA, B05001_001M, and B05001_001MA. `tidycensus` cleaned up all this detail for us, so we can focus on the variables that end in "E" for estimate. This table shows that variable B05001_001 corresponds to the total population. Variable B05001_002 is the number of U.S. citizens born in the United States in your county.

(10) Using the codes listed online as a reference, calculate the percentage of people who are not a U.S. citizen in your county.

In the last module, we did not pull a whole table into R. Some tables contain much more information than we need. If you just want a few variables, you can click the "selected variables" link next to a table of interest, and then import the only the variable codes you want.

(11) Modify the code below to import two or more variables for your county from any ACS table. Do not add "E", "EA", "M", or "MA" at the end of the variable codes in this function.

```r
pulling_variables <- get_acs(geography = "______",
                             state = "__",
                             county = "__________",
                             year = 2017,
                             survey = "acs5",
                             variables = c("B?????_???", "_______"))
# add more variables if you'd like to
```

Optional: (12) Pull data from the American Community Survey and calculate any metrics that you think might be changing in the farming community of your county. You can pull from different tables and different years, and use any coding methods you've learned. This is your time to experiment!

If you'd like more information about how you can use tidycensus, you can visit their tutorial site at https://walker-data.com/tidycensus/articles/basic-usage.html.

Census of Agriculture

There are a few packages similar to tidycensus built to pull data from the Census of Agriculture. Like tidycensus, you will need to request an API key and find the different data options online.

(13) Install the rnassqs package in the Console with install.packages("rnassqs").

(14) Read in the rnassqs library.

(15) Go to https://quickstats.nass.usda.gov/api/ and click "Request an API key" on the left side of the page to get an API key from the USDA.

(16) Import your API key with the code below. Hide your key with include=FALSE.

```
nassqs_auth(key = "YOUR API KEY HERE")
```

The general format of an rnassqs query is to specify a list of parameters that should apply to the data you want to pull, and then feed that list into the nassqs function. For example, to find the number of farms with conservation easements, I'll first specify a list of parameters that will get me the information I want. (17) Create this list of parameters for your county. I have added a new line for readability for the short_desc element, but you will need to replace it with a single space.

```
easements_params <- list(state_alpha = "ID",
                         agg_level_desc = "COUNTY",
                         county_name = "BEAR LAKE",
                         commodity_desc = "GOVT PROGRAMS",
                         short_desc = "GOVT PROGRAMS, FEDERAL,
                            CONSERVATION & WETLANDS - NUMBER OF OPERATIONS")
```

This will tell the rnassqs code to retrieve all data from the Census of Agriculture that matches these values.

(18) Pull the data from the Census of Agriculture by using the list of parameters we created as the argument for the function nassqs, which pulls data from the Census of Agriculture servers.

```
farms_with_conservation <- nassqs(easements_params,
                                   as_numeric = TRUE,
                                   progress_bar = FALSE)
```

(19) Print your data. There should be 5 rows, with information about the number of farms with conservation easements in 1997, 2002, 2007, 2012, and 2017. Even though the column names are different, the `Value` column still holds the value of interest.

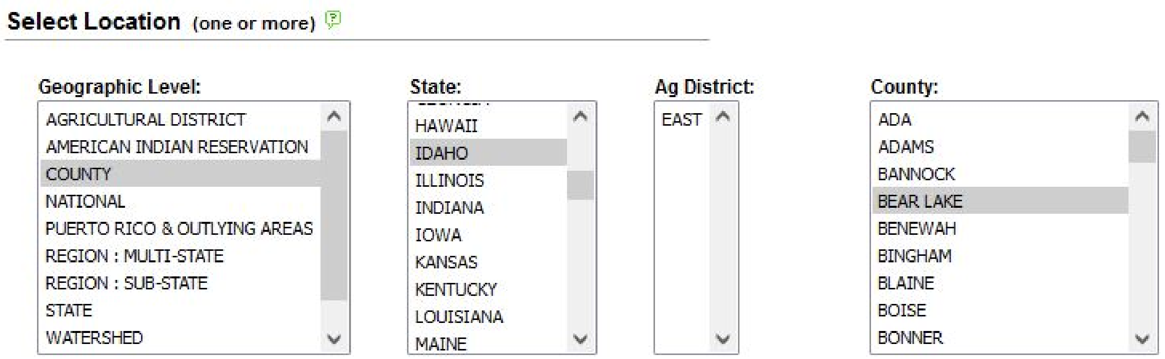
(20) Identify the number of farms with conservation easements in 2017 and **(21)** calculate the 20-year change in farms with easements from 1997 to 2017.

Optional: (22) Use `plot` to make a graph of the change in farms with easements over the 20 year period of 1997-2017. Are there any notable trends?

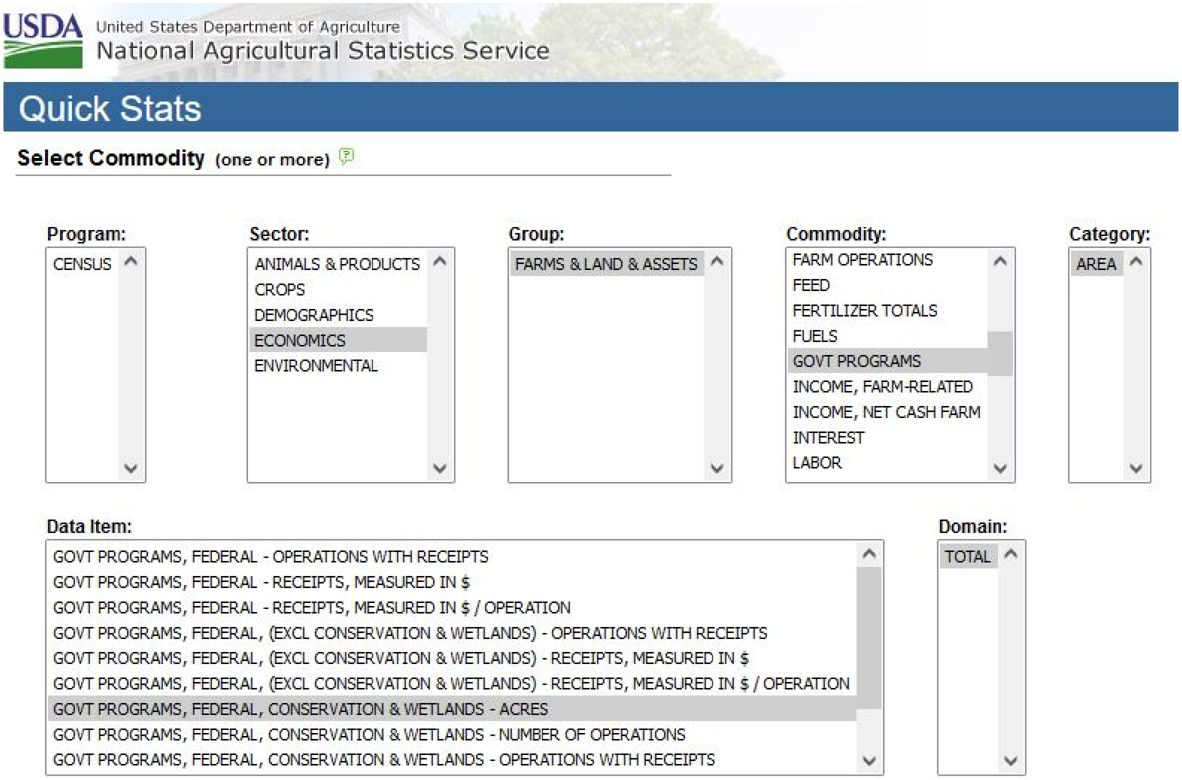
Finding your own parameters in the Census of Agriculture

You can explore all available parameters at https://quickstats.nass.usda.gov. Here, you can narrow down data selection to categories of information, locations, and years. After narrowing down your data selection, you can download a spreadsheet directly from the website. However, it is more reproducible to show the process you used to access data. This will help us track down the data for other counties if you find something interesting!

In Quick Stats, explore the different data categories. **(23)** First, select the "County" option in the "Geographic Level" box, then select "Idaho" in the "State" box and your county in the "County" box. This will make sure that any data options you select will be available for your county.



(24) Explore the different kinds of information you can access. Start by selecting an interesting "Sector," then continuing exploring and narrowing down options until you've identified one "Data Item" you'd like to analyze. Below is an example of the exploration I did to find the easements information:



USDA United States Department of Agriculture
National Agricultural Statistics Service

Quick Stats

Select Commodity (one or more)

Program:
CENSUS

Sector:
ANIMALS & PRODUCTS
CROPS
DEMOGRAPHICS
ECONOMICS
ENVIRONMENTAL

Group:
FARMS & LAND & ASSETS

Commodity:
FARM OPERATIONS
FEED
FERTILIZER TOTALS
FUELS
GOVT PROGRAMS
INCOME, FARM-RELATED
INCOME, NET CASH FARM
INTEREST
LABOR

Category:
AREA

Data Item:
GOVT PROGRAMS, FEDERAL - OPERATIONS WITH RECEIPTS
GOVT PROGRAMS, FEDERAL - RECEIPTS, MEASURED IN $
GOVT PROGRAMS, FEDERAL - RECEIPTS, MEASURED IN $ / OPERATION
GOVT PROGRAMS, FEDERAL, (EXCL CONSERVATION & WETLANDS) - OPERATIONS WITH RECEIPTS
GOVT PROGRAMS, FEDERAL, (EXCL CONSERVATION & WETLANDS) - RECEIPTS, MEASURED IN $
GOVT PROGRAMS, FEDERAL, (EXCL CONSERVATION & WETLANDS) - RECEIPTS, MEASURED IN $ / OPERATION
GOVT PROGRAMS, FEDERAL, CONSERVATION & WETLANDS - ACRES
GOVT PROGRAMS, FEDERAL, CONSERVATION & WETLANDS - NUMBER OF OPERATIONS
GOVT PROGRAMS, FEDERAL, CONSERVATION & WETLANDS - OPERATIONS WITH RECEIPTS

Domain:
TOTAL

To pull the data item you want, you will need to use the language of `rnassqs`. You can see a full description of every option by running `?nassqs` in the Console. **(25)** Use the values you selected in Quick Stats to make a list of parameters.

```
my_parameters <- list(state_alpha = "ID",
                      agg_level_desc = "COUNTY",
                      county_name = "YOUR COUNTY",
                      year = YYYY,
                      sector_desc = "", # Sector
                      group_desc = "", # Group
                      commodity_desc = "", # Commodity
                      statisticcat_desc = "", # Category
                      short_desc = "", # Data Item
                      domain_desc = "") # Domain
```

You don't need to use all of these options. For example, in step (17), I didn't use the `year` argument, so I got all the available years.

Optional: (26) Use the parameter list you created in step (25) in the function `nassqs` to pull the data that matches your parameters into `R`. Calculate any metrics that you think might be changing in the farming community of your county. You can pull from different parameters and different years, and use any coding methods you've learned. This is your time to experiment!

If you'd like more information about how you can use `rnassqs`, you can visit their tutorial site at https://cran.r-project.org/web/packages/rnassqs/vignettes/rnassqs.html.

Finishing up

(27) Summarize the information you learned about your county as a conclusion to this report.

(28) Go back through your report and add short explanations for what each code chunk does in your own words if you haven't done so already. **(29)** Render your report to a PDF and email it to [INSERT EMAIL HERE].

Statement of original and referenced work:

The entirety of this module is original work authored by Carolyn Koehn.

License

This module is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).