ACS Data Load

Alec Jasen

September 19, 2020

1 Introduction

This is the writeup for the ACS data loading exercise. The python file which executes the code is located in my github repo as `acs_data_load_full.py` since it does the download, transform/clean, upload process.

2 API

I chose to use the CensusData API available in python. I chose this for several reasons. My first attempt was downloading data from the census.data.gov site itself locally and `scp` the csv's to the EC2 instance. However, the site was being buggy and was stopping me from manual downloads. I next checked the Census Buruea API and they provide a RESTful API to get their data. On Slack someone mentioned a python API to accomplish the download task. Exploration showed the API makes those RESTful API calls for you, so that seemed the best way to go.

3 Variable Selection

I decided to grab the age by gender data at the block level for every county in Pennsylvania. If legislators represent their districts' constituents, then age and gender are known proxies for support of certain kinds of legislation. One specific application in mind is aggregating by district information from bill sponsors. Another is aggregating to the state level to see if age/gender breakdowns are useful proxies for looking at which kinds of legislation pass over time.

The data was broken up into finer groups that I thought might be useful, so I aggregated them at the level the Census chose to when making a voting graphic. The one they miss here, which isn't a voting constituent but which lawmakers write legislation for, is the under 18 category, so I put them together as well. This made for 10 data columns, 5 for each gender.

4 Other Details

I wrote a python script to encapsulate the entire process because only the psql call was not in python. I used subprocess to run the SQL file at the end. The SQL file was created by mirroring the instructions provided. The data has a few extra columns, since the details of each block were hidden in a `censusgeo` object. I parsed them out, reset the index, and added `state, county, tract,` and `block_group` as variables, meaning my personal table has 14. This should allow for higher levels of spatial aggregation as needed.