census_query_tutorial_v3

August 12, 2021

1 Python Tutorial Program: Gathering and Exporting Census Data

By Kenneth Burchfiel

You can find my blog post on this code at https://kburchfiel3.wordpress.com/2021/08/12/python-tutorial-program-retrieving-u-s-census-data/ .

This program demonstrates how Python (along with the census library, available at https://github.com/datamade/census) can be used to retrieve and export US Census data from thousands of zip codes. Although this tutorial program will focus on gathering education, family type, and income/poverty statistics from the American Community Survey, it should be a useful reference for those wishing to gather other types of census data instead.

Before being able to run the code below on your computer, you'll need to install the census libarry and obtain a free Census API key. See the above link for instructions.

First, I imported a number of libraries:

[31]:
```
import time
start_time = time.time() # Allows the program's runtime to be measured
from census import Census
# import us I didn't end up using this library, but you may find it useful for␣
 ↪your own Census query program. See https://github.com/datamade/census for␣
 ↪more information.
import pandas as pd
import numpy as np
import statsmodels.api as sm
```

Instead of hard coding the year into my Census queries, I chose to set it as a variable so that the queries could be modified more easily. I picked 2019 because it was the recent year that American Community Survey census data was available.

[32]:
```
year = 2019
```

Next, I imported my Census API key into the code. I stored the path to the key and the key itself in separate file locations.

[33]:
```
with open('..\\keys\\census_api_file_path.txt') as fin:
    api_key_path = fin.readline()
with open(api_key_path) as fin:
    api_key = fin.readline()
```

[34]:
```
c = Census(api_key) # See https://github.com/datamade/census
```

The next step was to locate the source of the data that I was interested in. For this program, I chose to retrieve zip code statistics for the following variables:

1. Household types (mostly married households vs. ones led by a female householder with no spouse present, which, for brevity's sake, I'll abbreviate as 'female-householder' homes.
2. The presence of children within these households
3. Median household income
4. Poverty status by family type
5. Poverty status by family type and the highest level of education completed

To search for this data, I used the Census's API site (https://api.census.gov/data.html). This is a very helpful site that provides links to different data sources, along with lists of groups and variables within those data sources.

For example, to access data from the 2019 American Community Survey, I searched in the above page in my web browser for 'acs5', then found the most recent year–which, in this case, happened to be 2019. To confirm that I could access data at the zip code level within this table, I could click on the 'geography' hyperlink (https://api.census.gov/data/2019/acs/acs5/geography.html). To figure out what types of data this survey provides, I clicked on its 'groups' hyperlink (https://api.census.gov/data/2019/acs/acs5/groups.html).

This groups page had 1,136 (!) different types of data that I could choose from. Fortunately, there were lots of options available for my variables of interest (marriage, income, education, household type, etc.)

The Census data site also provided an 'examples' page for accessing American Community Survey data (https://api.census.gov/data/2019/acs/acs5/examples.html), although the query format I used differed somewhat from the examples shown there.

I chose to query Census data in this program by: 1. Organizing different queries in dictionaries 2. Adding these dictionaries to a list (which I named 'metric_list') 3. Looping through this list 4. Storing the output of the queries in a DataFrame

The first two steps are shown below. I ended up adding many different queries to my dictionary, but you may choose to retrieve data for only a couple variables.

Each dictionary is based off information available on the Census Data page for a particular 'group.' For instance, to find data on the presence of children in households by household type, I chose to look into table B11005, 'HOUSEHOLDS BY PRESENCE OF PEOPLE UNDER 18 YEARS BY HOUSEHOLD TYPE' (which can be found on https://api.census.gov/data/2019/acs/acs5/groups.html). Clicking the 'selected variables' link for that group took me to https://api.census.gov/data/2019/acs/acs5/groups/B11005.html. This page shows all the different statistics available for the 'HOUSEHOLDS BY PRESENCE OF PEOPLE UNDER 18 YEARS BY HOUSEHOLD TYPE' group.

I stored the following information from these pages within the dictionaries below:

1. 'Name': the code on the Census website for that particular variable (e.g. B11005_001E).
2. 'Label': the Census's text description of that variable (e.g. 'Estimate!!Total:')
3. 'Concept': the Census's text description of the group to which the variable belongs (e.g. 'HOUSEHOLDS BY PRESENCE OF PEOPLE UNDER 18 YEARS BY HOUSEHOLD TYPE").

I also added an 'Alias' key to store my own description of these metrics. These aliases then served as column names in the Pandas DataFrame that stored the results of these queries. That DataFrame will appear later in this program.

I could have made the dictionaries simpler by including only the 'Name' and 'Alias' components, as the 'Label' and 'Concept' keys are neither used in the census queries nor displayed in the table. However, they can serve as a helpful reference for distinguishing between subtly different variable types.

[35]:
```
metric_list = []

# Group 1: Information on households by presence of children (see https://api.
 ↪census.gov/data/2019/acs/acs5/groups/B11005.html)

metric_list.append({'Name':'B11005_001E', 'Label':'Estimate!!Total:', 'Concept':
 ↪'HOUSEHOLDS BY PRESENCE OF PEOPLE UNDER 18 YEARS BY HOUSEHOLD TYPE','Alias':
 ↪'Households'})

metric_list.append({'Name':'B11005_013E', 'Label':'        Estimate!!Total:!!
 ↪Households with no people under 18 years:!!Family households:!!
 ↪Married-couple family', 'Concept':'HOUSEHOLDS BY PRESENCE OF PEOPLE UNDER 18␣
 ↪YEARS BY HOUSEHOLD TYPE','Alias':
 ↪'Married_couple_households_with_no_children'})

metric_list.append({'Name':'B11005_002E', 'Label':'Estimate!!Total:!!Households␣
 ↪with one or more people under 18 years:', 'Concept':'HOUSEHOLDS BY PRESENCE␣
 ↪OF PEOPLE UNDER 18 YEARS BY HOUSEHOLD TYPE:','Alias':
 ↪'Households_with_1_or_more_children'})

metric_list.append({'Name':'B11005_004E', 'Label':'Estimate!!Total:!!Households␣
 ↪with one or more people under 18 years:!!Family households:!!Married-couple␣
 ↪family', 'Concept':'HOUSEHOLDS BY PRESENCE OF PEOPLE UNDER 18 YEARS BY␣
 ↪HOUSEHOLD TYPE','Alias':'Married_couple_households_with_1_or_more_children'})

# Group 2: median household income
```

```
metric_list.append({'Name':'B19013_001E', 'Label':'Estimate!!Median household␣
 ↪income in the past 12 months (in YYYY inflation-adjusted dollars)',␣
 ↪'Concept':'MEDIAN HOUSEHOLD INCOME IN THE PAST 12 MONTHS (IN YYYY␣
 ↪INFLATION-ADJUSTED DOLLARS)','Alias':'Median_household_income'})

# Group 3: Numbers of children below/not below the poverty level in different␣
 ↪family types

metric_list.append({'Name':'B17006_002E', 'Label':'Estimate!!Total:!!Income in␣
 ↪the past 12 months below poverty level:', 'Concept':'POVERTY STATUS IN THE␣
 ↪PAST 12 MONTHS OF RELATED CHILDREN UNDER 18 YEARS BY FAMILY TYPE BY AGE OF␣
 ↪RELATED CHILDREN UNDER 18 YEARS','Alias':'Children_below_poverty_level'})

metric_list.append({'Name':'B17006_016E', 'Label':'Estimate!!Total:!!Income in␣
 ↪the past 12 months at or above poverty level:', 'Concept':'POVERTY STATUS IN␣
 ↪THE PAST 12 MONTHS OF RELATED CHILDREN UNDER 18 YEARS BY FAMILY TYPE BY AGE␣
 ↪OF RELATED CHILDREN UNDER 18 YEARS','Alias':
 ↪'Children_at_or_above_poverty_level'})

metric_list.append({'Name':'B17006_003E', 'Label':'Estimate!!Total:!!Income in␣
 ↪the past 12 months below poverty level:!!In married-couple family:',␣
 ↪'Concept':'POVERTY STATUS IN THE PAST 12 MONTHS OF RELATED CHILDREN UNDER 18␣
 ↪YEARS BY FAMILY TYPE BY AGE OF RELATED CHILDREN UNDER 18 YEARS','Alias':
 ↪'Children_in_married_couple_families_below_poverty_level'})

metric_list.append({'Name':'B17006_017E', 'Label':'Estimate!!Total:!!Income in␣
 ↪the past 12 months at or above poverty level:!!In married-couple family:',␣
 ↪'Concept':'POVERTY STATUS IN THE PAST 12 MONTHS OF RELATED CHILDREN UNDER 18␣
 ↪YEARS BY FAMILY TYPE BY AGE OF RELATED CHILDREN UNDER 18 YEARS','Alias':
 ↪'Children_in_married_couple_families_at_or_above_poverty_level'})

metric_list.append({'Name':'B17006_012E', 'Label':'Estimate!!Total:!!Income in␣
 ↪the past 12 months below poverty level:!!In other family:!!Female␣
 ↪householder, no spouse present:', 'Concept':'POVERTY STATUS IN THE PAST 12␣
 ↪MONTHS OF RELATED CHILDREN UNDER 18 YEARS BY FAMILY TYPE BY AGE OF RELATED␣
 ↪CHILDREN UNDER 18 YEARS','Alias':
 ↪'Children_in_female_householder_families_below_poverty_level'})

metric_list.append({'Name':'B17006_026E', 'Label':'        Estimate!!Total:!!
 ↪Income in the past 12 months at or above poverty level:!!In other family:!!
 ↪Female householder, no spouse present:', 'Concept':'POVERTY STATUS IN THE␣
 ↪PAST 12 MONTHS OF RELATED CHILDREN UNDER 18 YEARS BY FAMILY TYPE BY AGE OF␣
 ↪RELATED CHILDREN UNDER 18 YEARS','Alias':
 ↪'Children_in_female_householder_families_at_or_above_poverty_level'})
```

```
metric_list.append({'Name':'B17006_008E', 'Label':'Estimate!!Total:!!Income in␣
 ↪the past 12 months below poverty level:!!In other family:!!Male householder,␣
 ↪no spouse present:', 'Concept':'POVERTY STATUS IN THE PAST 12 MONTHS OF␣
 ↪RELATED CHILDREN UNDER 18 YEARS BY FAMILY TYPE BY AGE OF RELATED CHILDREN␣
 ↪UNDER 18 YEARS','Alias':
 ↪'Children_in_male_householder_families_below_poverty_level'})

metric_list.append({'Name':'B17006_022E', 'Label':'Estimate!!Total:!!Income in␣
 ↪the past 12 months at or above poverty level:!!In other family:!!Male␣
 ↪householder, no spouse present:', 'Concept':'POVERTY STATUS IN THE PAST 12␣
 ↪MONTHS OF RELATED CHILDREN UNDER 18 YEARS BY FAMILY TYPE BY AGE OF RELATED␣
 ↪CHILDREN UNDER 18 YEARS','Alias':
 ↪'Children_in_male_householder_families_at_or_above_poverty_level'})

# Group 4: poverty status by household type by householder's highest education␣
 ↪level

metric_list.append({'Name':'B17018_004E', 'Label':'Estimate!!Total:!!Income in␣
 ↪the past 12 months below poverty level:!!Married-couple family:!!Less than␣
 ↪high school graduate', 'Concept':'POVERTY STATUS IN THE PAST 12 MONTHS OF␣
 ↪FAMILIES BY HOUSEHOLD TYPE BY EDUCATIONAL ATTAINMENT OF HOUSEHOLDER','Alias':
 ↪'Number_of_married-couple_families_below_the_poverty_level_where_householder_did_not_gradua

metric_list.append({'Name':'B17018_021E', 'Label':'Estimate!!Total:!!Income in␣
 ↪the past 12 months at or above poverty level:!!Married-couple family:!!Less␣
 ↪than high school graduate', 'Concept':'POVERTY STATUS IN THE PAST 12 MONTHS␣
 ↪OF FAMILIES BY HOUSEHOLD TYPE BY EDUCATIONAL ATTAINMENT OF␣
 ↪HOUSEHOLDER','Alias':
 ↪'Number_of_married-couple_families_at_or_above_the_poverty_level_where_householder_did_not_

metric_list.append({'Name':'B17018_005E', 'Label':'Estimate!!Total:!!Income in␣
 ↪the past 12 months below poverty level:!!Married-couple family:!!High school␣
 ↪graduate (includes equivalency)', 'Concept':'POVERTY STATUS IN THE PAST 12␣
 ↪MONTHS OF FAMILIES BY HOUSEHOLD TYPE BY EDUCATIONAL ATTAINMENT OF␣
 ↪HOUSEHOLDER','Alias':
 ↪'Number_of_married-couple_families_below_the_poverty_level_where_householder\'s_highest_edu
 ↪equivalent'})

metric_list.append({'Name':'B17018_022E', 'Label':'Estimate!!Total:!!Income in␣
 ↪the past 12 months at or above poverty level:!!Married-couple family:!!High␣
 ↪school graduate (includes equivalency)', 'Concept':'POVERTY STATUS IN THE␣
 ↪PAST 12 MONTHS OF FAMILIES BY HOUSEHOLD TYPE BY EDUCATIONAL ATTAINMENT OF␣
 ↪HOUSEHOLDER','Alias':
 ↪'Number_of_married-couple_families_at_or_above_the_poverty_level_where_householder\'s_highe
 ↪equivalent'})
```

```
metric_list.append({'Name':'B17018_006E', 'Label':'Estimate!!Total:!!Income in␣
 ↪the past 12 months below poverty level:!!Married-couple family:!!Some␣
 ↪college, associate\'s degree', 'Concept':'POVERTY STATUS IN THE PAST 12␣
 ↪MONTHS OF FAMILIES BY HOUSEHOLD TYPE BY EDUCATIONAL ATTAINMENT OF␣
 ↪HOUSEHOLDER','Alias':
 ↪'Number_of_married-couple_families_below_the_poverty_level_where_householder\'s_highest_edu

metric_list.append({'Name':'B17018_023E', 'Label':'Estimate!!Total:!!Income in␣
 ↪the past 12 months at or above poverty level:!!Married-couple family:!!Some␣
 ↪college, associate\'s degree', 'Concept':'POVERTY STATUS IN THE PAST 12␣
 ↪MONTHS OF FAMILIES BY HOUSEHOLD TYPE BY EDUCATIONAL ATTAINMENT OF␣
 ↪HOUSEHOLDER','Alias':
 ↪'Number_of_married-couple_families_at_or_above_the_poverty_level_where_householder\'s_highe

metric_list.append({'Name':'B17018_007E', 'Label':'Estimate!!Total:!!Income in␣
 ↪the past 12 months below poverty level:!!Married-couple family:!!Bachelor\'s␣
 ↪degree or higher', 'Concept':'POVERTY STATUS IN THE PAST 12 MONTHS OF␣
 ↪FAMILIES BY HOUSEHOLD TYPE BY EDUCATIONAL ATTAINMENT OF HOUSEHOLDER','Alias':
 ↪'Number_of_married-couple_families_below_the_poverty_level_where_householder\'s_highest_edu

metric_list.append({'Name':'B17018_024E', 'Label':'Estimate!!Total:!!Income in␣
 ↪the past 12 months at or above poverty level:!!Married-couple family:!!
 ↪Bachelor\'s degree or higher', 'Concept':'POVERTY STATUS IN THE PAST 12␣
 ↪MONTHS OF FAMILIES BY HOUSEHOLD TYPE BY EDUCATIONAL ATTAINMENT OF␣
 ↪HOUSEHOLDER','Alias':
 ↪'Number_of_married-couple_families_at_or_above_the_poverty_level_where_householder\'s_highe

metric_list.append({'Name':'B17018_015E', 'Label':'Estimate!!Total:!!Income in␣
 ↪the past 12 months below poverty level:!!Other families:!!Female␣
 ↪householder, no spouse present:!!Less than high school graduate', 'Concept':
 ↪'POVERTY STATUS IN THE PAST 12 MONTHS OF FAMILIES BY HOUSEHOLD TYPE BY␣
 ↪EDUCATIONAL ATTAINMENT OF HOUSEHOLDER','Alias':
 ↪'Number_of_female-householder_families_below_the_poverty_level_where_householder_did_not_gr

metric_list.append({'Name':'B17018_032E', 'Label':'Estimate!!Total:!!Income in␣
 ↪the past 12 months at or above poverty level:!!Other families:!!Female␣
 ↪householder, no spouse present:!!Less than high school graduate', 'Concept':
 ↪'POVERTY STATUS IN THE PAST 12 MONTHS OF FAMILIES BY HOUSEHOLD TYPE BY␣
 ↪EDUCATIONAL ATTAINMENT OF HOUSEHOLDER','Alias':
 ↪'Number_of_female-householder_families_at_or_above_the_poverty_level_where_householder_did_
```

```
metric_list.append({'Name':'B17018_016E', 'Label':'Estimate!!Total:!!Income in␣
 ↪the past 12 months below poverty level:!!Other families:!!Female␣
 ↪householder, no spouse present:!!High school graduate (includes␣
 ↪equivalency)', 'Concept':'POVERTY STATUS IN THE PAST 12 MONTHS OF FAMILIES␣
 ↪BY HOUSEHOLD TYPE BY EDUCATIONAL ATTAINMENT OF HOUSEHOLDER','Alias':
 ↪'Number_of_female-householder_families_below_the_poverty_level_where_householder\'s_highest
 ↪equivalent'})

metric_list.append({'Name':'B17018_033E', 'Label':'Estimate!!Total:!!Income in␣
 ↪the past 12 months at or above poverty level:!!Other families:!!Female␣
 ↪householder, no spouse present:!!High school graduate (includes␣
 ↪equivalency)', 'Concept':'POVERTY STATUS IN THE PAST 12 MONTHS OF FAMILIES␣
 ↪BY HOUSEHOLD TYPE BY EDUCATIONAL ATTAINMENT OF HOUSEHOLDER','Alias':
 ↪'Number_of_female-householder_families_at_or_above_the_poverty_level_where_householder\'s_h
 ↪equivalent'})

metric_list.append({'Name':'B17018_017E', 'Label':'Estimate!!Total:!!Income in␣
 ↪the past 12 months below poverty level:!!Other families:!!Female␣
 ↪householder, no spouse present:!!Some college, associate\'s degree',␣
 ↪'Concept':'POVERTY STATUS IN THE PAST 12 MONTHS OF FAMILIES BY HOUSEHOLD␣
 ↪TYPE BY EDUCATIONAL ATTAINMENT OF HOUSEHOLDER','Alias':
 ↪'Number_of_female-householder_families_below_the_poverty_level_where_householder\'s_highest

metric_list.append({'Name':'B17018_034E', 'Label':'Estimate!!Total:!!Income in␣
 ↪the past 12 months at or above poverty level:!!Other families:!!Female␣
 ↪householder, no spouse present:!!Some college, associate\'s degree',␣
 ↪'Concept':'POVERTY STATUS IN THE PAST 12 MONTHS OF FAMILIES BY HOUSEHOLD␣
 ↪TYPE BY EDUCATIONAL ATTAINMENT OF HOUSEHOLDER','Alias':
 ↪'Number_of_female-householder_families_at_or_above_the_poverty_level_where_householder\'s_h

metric_list.append({'Name':'B17018_018E', 'Label':'Estimate!!Total:!!Income in␣
 ↪the past 12 months below poverty level:!!Other families:!!Female␣
 ↪householder, no spouse present:!!Bachelor\'s degree or higher', 'Concept':
 ↪'POVERTY STATUS IN THE PAST 12 MONTHS OF FAMILIES BY HOUSEHOLD TYPE BY␣
 ↪EDUCATIONAL ATTAINMENT OF HOUSEHOLDER','Alias':
 ↪'Number_of_female-householder_families_below_the_poverty_level_where_householder\'s_highest

metric_list.append({'Name':'B17018_035E', 'Label':'Estimate!!Total:!!Income in␣
 ↪the past 12 months at or above poverty level:!!Other families:!!Female␣
 ↪householder, no spouse present:!!Bachelor\'s degree or higher', 'Concept':
 ↪'POVERTY STATUS IN THE PAST 12 MONTHS OF FAMILIES BY HOUSEHOLD TYPE BY␣
 ↪EDUCATIONAL ATTAINMENT OF HOUSEHOLDER','Alias':
 ↪'Number_of_female-householder_families_at_or_above_the_poverty_level_where_householder\'s_h


# Group 5: Highest education level completed
```

```
metric_list.append({'Name':'B16010_001E', 'Label':'Estimate!!Total:', 'Concept':
 ↪'EDUCATIONAL ATTAINMENT AND EMPLOYMENT STATUS BY LANGUAGE SPOKEN AT HOME FOR␣
 ↪THE POPULATION 25 YEARS AND OVER','Alias':'Total_number_of_individuals_25+y/
 ↪o_in_table_B16010'})

metric_list.append({'Name':'B16010_002E', 'Label':'        Estimate!!Total:!!
 ↪Less than high school graduate:', 'Concept':'EDUCATIONAL ATTAINMENT AND␣
 ↪EMPLOYMENT STATUS BY LANGUAGE SPOKEN AT HOME FOR THE POPULATION 25 YEARS AND␣
 ↪OVER','Alias':'Number_of_individuals_25+y/
 ↪o_who_did_not_graduate_high_school'})

metric_list.append({'Name':'B16010_015E', 'Label':'Estimate!!Total:!!High␣
 ↪school graduate (includes equivalency):', 'Concept':'EDUCATIONAL ATTAINMENT␣
 ↪AND EMPLOYMENT STATUS BY LANGUAGE SPOKEN AT HOME FOR THE POPULATION 25 YEARS␣
 ↪AND OVER','Alias':'Number_of_individuals_25+y/
 ↪o_whose_highest_education_level_=_high_school_graduate/equivalent'})

metric_list.append({'Name':'B16010_028E', 'Label':'Estimate!!Total:!!
 ↪some_college_or_associate\'s degree:', 'Concept':'EDUCATIONAL ATTAINMENT AND␣
 ↪EMPLOYMENT STATUS BY LANGUAGE SPOKEN AT HOME FOR THE POPULATION 25 YEARS AND␣
 ↪OVER','Alias':'Number_of_individuals_25+_y/
 ↪o_whose_highest_education_level_=_some_college/associate\'s_degree'})

metric_list.append({'Name':'B16010_041E', 'Label':'Estimate!!Total:!!
 ↪Bachelor\'s degree or higher:', 'Concept':'EDUCATIONAL ATTAINMENT AND␣
 ↪EMPLOYMENT STATUS BY LANGUAGE SPOKEN AT HOME FOR THE POPULATION 25 YEARS AND␣
 ↪OVER','Alias':'Number_of_individuals_25+_y/
 ↪o_whose_highest_education_level_=_bachelor\'s_degree_or_higher'})



# metric_list.append({'Name':'', 'Label':'', 'Concept':'','Alias':''}) #␣
 ↪Template for additional dictioanries
```

[36]:
```
len(metric_list) # Shows the number of queries to be processed by the Census API
```

[36]:
```
34
```

The following code provides an example of how the census library works. It derives from the examples shown at https://github.com/datamade/census, but differs in that the query is returned for a particular zip code rather than for a particular state.

The code following 'NAME' in this example is one of the variable codes entered into the dictionary list above.

[37]:
```
sample_year = 2019
sample_zip = 10940
```

```
sample_query = c.acs5.get(('NAME', 'B17018_003E'), {'for': 'zip code tabulation␣
↪area:{}'.format(sample_zip), 'in':'state:36', 'year':sample_year}) # 36 is␣
↪the FIPS code for New York. This can be looked up online; alternately, you␣
↪can find it using the us library referenced in https://github.com/datamade/
↪census.
# the {}.format() convention allows a variable to be passed into the 'zip code␣
↪tabulation area:' string. For an overview of this method, see https://docs.
↪python.org/3/library/string.html#formatstrings.
sample_query
```

[37]:
```
[{'NAME': 'ZCTA5 10940',
  'B17018_003E': 551.0,
  'state': '36',
  'zip code tabulation area': '10940'}]
```

The output of the above query is a list of dictionaries. Since only one zip code was entered, the length of this list is 1, but it will be over 33,000 in the for loop below. The number following B17018_003E is the value represented by that code (in this case, the number of married-couple families below the poverty level) for the sample zip code.

The for loop below is the heart of this program. It performs census queries using the codes in the dictionaries within metric_list for all available zip codes, then converts the results into a Pandas DataFrame using list comprehensions and the pd.merge() function.

[8]:
```
for i in range(len(metric_list)):
    census_query = c.acs5.get(('NAME', metric_list[i]['Name']), {'for': 'zip␣
↪code tabulation area:*', 'in':'state:*', 'year':year})
    #metric_list[i]['Name'] returns the variable code (e.g. B17006_012E) that␣
↪the get() function will process.
    # The asterisks after zip code tabulation area and state instruct the get()␣
↪function to return results for all zip codes in all states. As shown in the␣
↪earlier example, however, you can also select results from one particular␣
↪zip code and/or state.
    zip_list = [census_query[j]['NAME'][-5:] for j in range(len(census_query))]␣
↪# Refer back to the output from the sample query. the 'NAME' component of␣
↪census_query equals 'ZCTA5' plus a 5-digit zip code. Therefore, [-5:] is␣
↪used to select only the 5-digit zip code.
    # zip_list, along with state_list and metric, use list comprehensions. See␣
↪https://docs.python.org/3/tutorial/datastructures.html and the last example␣
↪shown on https://pandas.pydata.org/pandas-docs/version/0.22/generated/pandas.
↪DataFrame.append.html .
    # j is used here to distinguish the range of zip codes shown in the list␣
↪comprehension from the range of queries shown in the above for loop.
    state_list = [census_query[j]['state'] for j in range(len(census_query))] #␣
↪Retrieves the state FIPS code from the output
```

```
    metric = [census_query[j][metric_list[i]['Name']] for j in␣
↪range(len(census_query))] # census_query[j] retrieves a query result for a␣
↪particular zip code. As shown in the sample query, the key for the metric␣
↪equals the code for the current query. That code can be accessed through␣
↪metric_list[i]['Name'].
    df_metric = pd.DataFrame(data={'Zip':zip_list, 'State':state_list,␣
↪metric_list[i]['Alias']:metric}) # This line creates a dictionary with 3␣
↪key-value pairs: (1) 'Zip' and zip_list; (2) 'State' and state_list; and (3)␣
↪the 'Alias' value within metric_list for the current query and metric. Note␣
↪that zip_list, state_list, and metric are all lists. This new dictionary is␣
↪then converted to the DataFrame df_metric using pd.DataFrame.

    if i == 0:
        df_results = df_metric # if the above instance of df_metric was the␣
↪first to be created, it will serve as the basis for df_results.
    # df_list.append(df_test)
    else:
        df_results = df_results.merge(right=df_metric,how='outer') # Further␣
↪instances of df_metric are added to the right of df_results using pd.merge.␣
↪As a result, each 'Alias' value for each query will become a column label␣
↪within df_results for its corresponding query.

df_results
```

[8]:

	Zip	State	Households	Married_couple_households_with_no_children	Households_with_1_or_more_children
0	53563	55	4282.0	1542.0	1448.0
1	53588	55	1782.0	714.0	377.0
2	53589	55	7925.0	2512.0	2415.0
3	53713	55	10129.0	1169.0	3274.0
4	53717	55	6127.0	1773.0	1568.0
…	…	…	…	…	…
33115	15201	42	6194.0	1254.0	748.0
33116	15636	42	1423.0	591.0	487.0
33117	15696	42	188.0	59.0	32.0
33118	15035	42	987.0	274.0	
33119	15222	42	2526.0	522.0	

```
33118                                208.0
33119                                 81.0

       Married_couple_households_with_1_or_more_children  \
0                                                1039.0
1                                                 289.0
2                                                1825.0
3                                                1255.0
4                                                1202.0
…                                                 …
33115                                             441.0
33116                                             455.0
33117                                              24.0
33118                                             144.0
33119                                              68.0

       Median_household_income  Children_below_poverty_level  \
0                      74518.0                          94.0
1                      61898.0                          56.0
2                      70585.0                         160.0
3                      42094.0                        1981.0
4                      80125.0                         408.0
…                        …                              …
33115                  61642.0                         150.0
33116                 118264.0                          36.0
33117                  60625.0                           0.0
33118                  45292.0                          46.0
33119                  99419.0                           0.0

       Children_at_or_above_poverty_level  \
0                                  2700.0
1                                   660.0
2                                  4363.0
3                                  3988.0
4                                  2055.0
…                                   …
33115                              1076.0
33116                               875.0
33117                                52.0
33118                               358.0
33119                               137.0

       Children_in_married_couple_families_below_poverty_level  …  \
0                                                   54.0        …
1                                                   15.0        …
2                                                   52.0        …
3                                                  857.0        …
```

```
4                                                    185.0        …
…                                                    …        …
33115                                                 42.0        …
33116                                                  0.0        …
33117                                                  0.0        …
33118                                                  0.0        …
33119                                                  0.0        …

       Number_of_female-householder_families_at_or_above_the_poverty_level_where
_householder's_highest_education_=_high_school_graduate/equivalent  \
0                                                     48.0
1                                                     17.0
2                                                    279.0
3                                                    255.0
4                                                     56.0
…                                                    …
33115                                                164.0
33116                                                 16.0
33117                                                  3.0
33118                                                 24.0
33119                                                  0.0

       Number_of_female-householder_families_below_the_poverty_level_where_house
holder's_highest_education_=_some_college_or_associate's_degree  \
0                                                     10.0
1                                                     12.0
2                                                     35.0
3                                                    179.0
4                                                      9.0
…                                                    …
33115                                                 57.0
33116                                                  0.0
33117                                                  0.0
33118                                                  8.0
33119                                                  0.0

       Number_of_female-householder_families_at_or_above_the_poverty_level_where
_householder's_highest_education_=_some_college_or_associate's_degree  \
0                                                    174.0
1                                                     22.0
2                                                    315.0
3                                                    572.0
4                                                     29.0
…                                                    …
33115                                                176.0
33116                                                  0.0
33117                                                  0.0
```

```
33118                                                 31.0
33119                                                  0.0

       Number_of_female-householder_families_below_the_poverty_level_where_house
holder's_highest_education_level_=_bachelor's_degree_or_higher  \
0                                                     10.0
1                                                      6.0
2                                                      5.0
3                                                      0.0
4                                                     60.0
…                                                     …
33115                                                  0.0
33116                                                  0.0
33117                                                  0.0
33118                                                  0.0
33119                                                  0.0

       Number_of_female-householder_families_at_or_above_the_poverty_level_where
_householder's_highest_education_level_=_bachelor's_degree_or_higher  \
0                                                     46.0
1                                                     29.0
2                                                    152.0
3                                                    217.0
4                                                    240.0
…                                                    …
33115                                                223.0
33116                                                  0.0
33117                                                  5.0
33118                                                  8.0
33119                                                 34.0

       Total_number_of_individuals_25+y/o_in_table_B16010  \
0                                                   7533.0
1                                                   3141.0
2                                                  14104.0
3                                                  13611.0
4                                                   9429.0
…                                                    …
33115                                               9961.0
33116                                               2758.0
33117                                                235.0
33118                                               1572.0
33119                                               3453.0

       Number_of_individuals_25+y/o_who_did_not_graduate_high_school  \
0                                                    497.0
1                                                    211.0
```

```
2                                                807.0
3                                               1797.0
4                                                139.0
…                                                …
33115                                            548.0
33116                                             69.0
33117                                             14.0
33118                                             91.0
33119                                             79.0

       Number_of_individuals_25+y/o_whose_highest_education_level_=_high_school_
graduate/equivalent  \
0                                               2500.0
1                                                935.0
2                                               3873.0
3                                               3160.0
4                                                752.0
…                                                …
33115                                           2056.0
33116                                            837.0
33117                                            115.0
33118                                            674.0
33119                                            242.0

       Number_of_individuals_25+_y/o_whose_highest_education_level_=_some_colleg
e/associate's_degree  \
0                                               2398.0
1                                               1027.0
2                                               4654.0
3                                               4385.0
4                                               1537.0
…                                                …
33115                                           2050.0
33116                                            511.0
33117                                             63.0
33118                                            561.0
33119                                            506.0

       Number_of_individuals_25+_y/o_whose_highest_education_level_=_bachelor's_
degree_or_higher
0                                               2138.0
1                                                968.0
2                                               4770.0
3                                               4269.0
4                                               7001.0
…                                                …
33115                                           5307.0
```

```
33116                                                    1341.0
33117                                                      43.0
33118                                                     246.0
33119                                                    2626.0

[33120 rows x 36 columns]
```

I admit that many of the column names are obscenely long and unwieldy. This is less of an issue when viewing the table as a CSV export (which I'll perform later), since spreadsheet software can make the columns a uniform width while allowing the full name to be displayed in a separate box. An alternative to these long names, though, would be to substitute in shorter names, then include a key explaining each of their meanings.

The following block of code reports the number of census datapoints that df_results currently contains. (Some row-column pairs are empty, so df.count() is used to determine the number of cells that do contain data.)

[9]:
```
cell_counts = df_results.count() # Creates a series that contains the number of␣
 ↪cells with data in each column.
total_count = 0
print(len(cell_counts))
for i in range(2, len(cell_counts)): # For loop starts at 2 to exclude the␣
 ↪first two columns, which merely contain zip and state information.
    total_count += cell_counts[i]
print("There are",'{:,}'.format(total_count),"cells with census data in␣
 ↪df_results so far.") # See https://docs.python.org/3/library/string.
 ↪html#formatstrings for an overview of the .format() function
```

```
36
There are 1,125,030 cells with census data in df_results so far.
```

So far, the values shown in the DataFrame are nominal in nature. For example, the table reports on the number of married-couple households with one or more children, but doesn't say what *proportion* have at laest one child–which is much more useful when comparing different zip codes.

Therefore, in the following code block, I added additional columns to the DataFrame that generate various proportions. Some of these were generated using pre-existing totals as a denominator, whereas others used the sum of two diferent statistics as the denominator. (For example, to calculate the proportion of children below the poverty level for a given zip code, I divided the number of children below the poverty level by the sum of (1) children below the poverty level and (2) children above the poverty level. This was a useful strategy when a given Census table didn't have a 'totals' row.

(When creating proportions, be careful about using a total in one table as the denominator for a proportion calculation that involves a separate table. For example, if Table A says that there are 10,000 kids in a zip code, and Table B says that there are 2,000 kids below the poverty line, you may be tempted to conclude that the proportion of children below the poverty line equals 2,000/10,000 = 0.2. However, suppose not all the kids identified in Table A show up in Table B, and that Table B doesn't have a totals row. In that case, you'd want to divide the proportion of kids in Table B above below the poverty level (2,000) by the number in Table B above the poverty level (let's say

it's 6,000) to arrive at a more accurate proportion–in this case, 2,000/(2,000+6,000) = 2,000/8,000 = 25%.)

[10]:
```
df_results['Married_couple_households_with_one_or_more_children_as_proportion_of_all_household
 ↪= df_results['Married_couple_households_with_1_or_more_children']/
 ↪df_results['Households']

df_results['Married_couple_households_with_one_or_more_children_as_proportion_of_all_household
 ↪= df_results['Married_couple_households_with_1_or_more_children']/
 ↪df_results['Households_with_1_or_more_children']

df_results['Proportion_of_children_below_poverty_level'] =␣
 ↪df_results['Children_below_poverty_level']/
 ↪(df_results['Children_below_poverty_level'] +␣
 ↪df_results['Children_at_or_above_poverty_level'])

df_results['Proportion_of_children_in_married_couple_families_below_poverty_level']␣
 ↪= df_results['Children_in_married_couple_families_below_poverty_level']/
 ↪(df_results['Children_in_married_couple_families_below_poverty_level'] +␣
 ↪df_results['Children_in_married_couple_families_at_or_above_poverty_level'])

df_results['Proportion_of_children_in_female_householder_families_below_poverty_level']␣
 ↪= df_results['Children_in_female_householder_families_below_poverty_level']/
 ↪(df_results['Children_in_female_householder_families_below_poverty_level'] +␣
 ↪df_results['Children_in_female_householder_families_at_or_above_poverty_level'])

df_results['Proportion_of_children_in_male_householder_families_below_poverty_level']␣
 ↪= df_results['Children_in_male_householder_families_below_poverty_level']/
 ↪(df_results['Children_in_male_householder_families_below_poverty_level'] +␣
 ↪df_results['Children_in_male_householder_families_at_or_above_poverty_level'])

# Calculating proportions of residents living below the poverty level by␣
 ↪education and household type

df_results['Proportion_of_married-couple_families_below_the_poverty_level_where_householder_di
 ↪=␣
 ↪df_results['Number_of_married-couple_families_below_the_poverty_level_where_householder_did
 ↪(df_results['Number_of_married-couple_families_below_the_poverty_level_where_householder_di

df_results['Proportion_of_married-couple_families_below_the_poverty_level_where_householder\'s
 ↪equivalent'] =␣
 ↪df_results['Number_of_married-couple_families_below_the_poverty_level_where_householder\'s_
 ↪equivalent']/
 ↪(df_results['Number_of_married-couple_families_below_the_poverty_level_where_householder\'s
 ↪equivalent']+df_results['Number_of_married-couple_families_at_or_above_the_poverty_level_wh
 ↪equivalent'])
```

```
df_results['Proportion_of_married-couple_families_below_the_poverty_level_where_householder\'s
 ↪=␣
 ↪df_results['Number_of_married-couple_families_below_the_poverty_level_where_householder\'s_
 ↪(df_results['Number_of_married-couple_families_below_the_poverty_level_where_householder\'s

df_results['Proportion_of_married-couple_families_below_the_poverty_level_where_householder\'s
 ↪=␣
 ↪df_results['Number_of_married-couple_families_below_the_poverty_level_where_householder\'s_
 ↪(df_results['Number_of_married-couple_families_below_the_poverty_level_where_householder\'s


df_results['Proportion_of_female-householder_families_below_the_poverty_level_where_householde
 ↪=␣
 ↪df_results['Number_of_female-householder_families_below_the_poverty_level_where_householder
 ↪(df_results['Number_of_female-householder_families_below_the_poverty_level_where_householde

df_results['Proportion_of_female-householder_families_below_the_poverty_level_where_householde
 ↪equivalent'] =␣
 ↪df_results['Number_of_female-householder_families_below_the_poverty_level_where_householder
 ↪equivalent']/
 ↪(df_results['Number_of_female-householder_families_below_the_poverty_level_where_householde
 ↪equivalent']+df_results['Number_of_female-householder_families_at_or_above_the_poverty_leve
 ↪equivalent'])

df_results['Proportion_of_female-householder_families_below_the_poverty_level_where_householde
 ↪=␣
 ↪df_results['Number_of_female-householder_families_below_the_poverty_level_where_householder
 ↪(df_results['Number_of_female-householder_families_below_the_poverty_level_where_householde

df_results['Proportion_of_female-householder_families_below_the_poverty_level_where_householde
 ↪=␣
 ↪df_results['Number_of_female-householder_families_below_the_poverty_level_where_householder
 ↪(df_results['Number_of_female-householder_families_below_the_poverty_level_where_householde

df_results['Proportion_of_individuals_25+y/o_who_did_not_graduate_high_school']␣
 ↪= df_results['Number_of_individuals_25+y/
 ↪o_who_did_not_graduate_high_school']/
 ↪(df_results['Total_number_of_individuals_25+y/o_in_table_B16010'])

df_results['Proportion_of_individuals_25+y/
 ↪o_whose_highest_education_level_=_high_school_graduate/equivalent'] =␣
 ↪df_results['Number_of_individuals_25+y/
 ↪o_whose_highest_education_level_=_high_school_graduate/equivalent']/
 ↪(df_results['Total_number_of_individuals_25+y/o_in_table_B16010'])
```

```
df_results['Proportion_of_individuals_25+_y/
 ↪o_whose_highest_education_level_=_some_college/associate\'s_degree'] =␣
 ↪df_results['Number_of_individuals_25+_y/
 ↪o_whose_highest_education_level_=_some_college/associate\'s_degree']/
 ↪(df_results['Total_number_of_individuals_25+y/o_in_table_B16010'])

df_results['Proportion_of_individuals_25+_y/
 ↪o_whose_highest_education_level_=_bachelor\'s_degree_or_higher'] =␣
 ↪df_results['Number_of_individuals_25+_y/
 ↪o_whose_highest_education_level_=_bachelor\'s_degree_or_higher']/
 ↪(df_results['Total_number_of_individuals_25+y/o_in_table_B16010'])


# df_results[''] = df_results['']/(df_results['']+df_results[''])


df_results.
 ↪sort_values('Married_couple_households_with_one_or_more_children_as_proportion_of_all_housel
df_results.reset_index(drop=True,inplace=True)
```

Here's how df_results looks with the additional proportions columns added in:

[11]:
```
df_results
```

[11]:

	Zip	State	Households	Married_couple_households_with_no_children	\
0	84743	49	1.0	0.0	
1	84540	49	24.0	0.0	
2	21862	24	12.0	0.0	
3	78802	48	79.0	0.0	
4	69365	31	9.0	0.0	
…	…	…	…	…	
33115	82715	56	0.0	0.0	
33116	83874	16	0.0	0.0	
33117	88417	35	0.0	0.0	
33118	76523	48	0.0	0.0	
33119	76874	48	0.0	0.0	

	Households_with_1_or_more_children	\
0	1.0	
1	24.0	
2	12.0	
3	79.0	
4	9.0	
…	…	
33115	0.0	
33116	0.0	
33117	0.0	

```
33118                                0.0
33119                                0.0

       Married_couple_households_with_1_or_more_children  \
0                                                    1.0
1                                                   24.0
2                                                   12.0
3                                                   79.0
4                                                    9.0
…                                                    …
33115                                                0.0
33116                                                0.0
33117                                                0.0
33118                                                0.0
33119                                                0.0

       Median_household_income  Children_below_poverty_level  \
0                -666666666.0                           0.0
1                -666666666.0                           0.0
2                -666666666.0                           0.0
3                -666666666.0                           0.0
4                -666666666.0                           0.0
…                        …                              …
33115            -666666666.0                           0.0
33116            -666666666.0                           0.0
33117            -666666666.0                           0.0
33118            -666666666.0                           0.0
33119            -666666666.0                           0.0

       Children_at_or_above_poverty_level  \
0                                     1.0
1                                    35.0
2                                    29.0
3                                    56.0
4                                    17.0
…                                    …
33115                                 0.0
33116                                 0.0
33117                                 0.0
33118                                 0.0
33119                                 0.0

       Children_in_married_couple_families_below_poverty_level  …  \
0                                                    0.0        …
1                                                    0.0        …
2                                                    0.0        …
3                                                    0.0        …
```

```
4                                                    0.0        …
…                                                    …          …
33115                                                0.0        …
33116                                                0.0        …
33117                                                0.0        …
33118                                                0.0        …
33119                                                0.0        …

       Proportion_of_married-couple_families_below_the_poverty_level_where_house
holder's_highest_education_=_some_college_or_associate's_degree  \
0                                                    NaN
1                                                    0.0
2                                                    NaN
3                                                    0.0
4                                                    NaN
…                                                    …
33115                                                NaN
33116                                                NaN
33117                                                NaN
33118                                                NaN
33119                                                NaN

       Proportion_of_married-couple_families_below_the_poverty_level_where_house
holder's_highest_education_level_=_bachelor's_degree_or_higher  \
0                                                    0.0
1                                                    NaN
2                                                    NaN
3                                                    NaN
4                                                    0.0
…                                                    …
33115                                                NaN
33116                                                NaN
33117                                                NaN
33118                                                NaN
33119                                                NaN

       Proportion_of_female-householder_families_below_the_poverty_level_where_h
ouseholder_did_not_graduate_high_school  \
0                                                    NaN
1                                                    NaN
2                                                    NaN
3                                                    NaN
4                                                    NaN
…                                                    …
33115                                                NaN
33116                                                NaN
33117                                                NaN
```

```
33118                                                 NaN
33119                                                 NaN

       Proportion_of_female-householder_families_below_the_poverty_level_where_h
ouseholder's_highest_education_=_high_school_graduate/equivalent  \
0                                                     NaN
1                                                     NaN
2                                                     NaN
3                                                     NaN
4                                                     NaN
…                                                     …
33115                                                 NaN
33116                                                 NaN
33117                                                 NaN
33118                                                 NaN
33119                                                 NaN

       Proportion_of_female-householder_families_below_the_poverty_level_where_h
ouseholder's_highest_education_=_some_college_or_associate's_degree  \
0                                                     NaN
1                                                     NaN
2                                                     NaN
3                                                     NaN
4                                                     NaN
…                                                     …
33115                                                 NaN
33116                                                 NaN
33117                                                 NaN
33118                                                 NaN
33119                                                 NaN

       Proportion_of_female-householder_families_below_the_poverty_level_where_h
ouseholder's_highest_education_level_=_bachelor's_degree_or_higher  \
0                                                     NaN
1                                                     NaN
2                                                     NaN
3                                                     NaN
4                                                     NaN
…                                                     …
33115                                                 NaN
33116                                                 NaN
33117                                                 NaN
33118                                                 NaN
33119                                                 NaN

       Proportion_of_individuals_25+y/o_who_did_not_graduate_high_school  \
0                                                     0.0
```

```
1                                                    0.0
2                                                    0.0
3                                                    0.0
4                                                    0.0
…                                                    …
33115                                                NaN
33116                                                NaN
33117                                                NaN
33118                                                NaN
33119                                                NaN

       Proportion_of_individuals_25+y/o_whose_highest_education_level_=_high_sch
ool_graduate/equivalent  \
0                                                0.812500
1                                                0.555556
2                                                1.000000
3                                                0.000000
4                                                0.571429
…                                                   …
33115                                               NaN
33116                                               NaN
33117                                               NaN
33118                                               NaN
33119                                               NaN

       Proportion_of_individuals_25+_y/o_whose_highest_education_level_=_some_co
llege/associate's_degree  \
0                                                0.125000
1                                                0.444444
2                                                0.000000
3                                                0.560284
4                                                0.000000
…                                                   …
33115                                               NaN
33116                                               NaN
33117                                               NaN
33118                                               NaN
33119                                               NaN

       Proportion_of_individuals_25+_y/o_whose_highest_education_level_=_bachelo
r's_degree_or_higher
0                                                0.062500
1                                                0.000000
2                                                0.000000
3                                                0.439716
4                                                0.428571
…                                                   …
```

```
33115                                                    NaN
33116                                                    NaN
33117                                                    NaN
33118                                                    NaN
33119                                                    NaN

[33120 rows x 54 columns]
```

A look at the first few rows in this table reveals that some median household income values are clearly inaccurate! $-666,666,666 is *not* the actual median household income in any zip code, yet that's the value listed for 2,194 counties, as shown below:

[12]:
```
len(df_results.query("Median_household_income == -666666666"))
```

[12]:
```
2194
```

This means that, when performing average calculations across the entire dataset, you must be extremely careful–otherwise, you'll end up with results like the one below:

[13]:
```
np.mean(df_results['Median_household_income'])
```

[13]:
```
-44154242.870303765
```

These results are, of course, skewed by the thousands of -666,666,666 values. The U.S. would be in dire shape if the average median household income among zip codes was truly $-44,154,242!

I then exported two versions of this DataFrame to a CSV. The first version (df_results_1k_plus_households) only includes zip codes with at least 1,000 households, since lower sample sizes in smaller zip codes can skew the sample sizes shown. The second version contains all zip codes present in the dataset.

[14]:
```
df_results_1k_plus_households = df_results.query("Households > 1000")
df_results_1k_plus_households.to_csv('census_query_results_1k_plus_households.
 ↪csv')
df_results.to_csv('census_query_results.csv')
```

As shown below, running the same average median household calculation on the reduced dataset produces a more accurate-looking number. Nevertheless, it would still be better to look through the DataFrame beforehand and perform any necessary data cleaning.

[15]:
```
np.mean(df_results_1k_plus_households['Median_household_income'])
```

[15]:
```
63212.38889870956
```

That concludes the main part of this tutorial program. I hope that you find these examples useful in performing your own census data analysis!

These census DataFrames can also be a great source of information for regression analyses. The following code blocks show how one of the DataFrames can be modified to serve as a data source for regressions (albeit without any data cleaning or checking). In the future, I may move these

regressions over to a separate tutorial program and provide detailed explanations of the code. In the meantime, I've left the code in place and added some brief explanations.

The first regression examined the relationship between poverty rates and whether children were in a married-couple family as opposed to a female-householder one. This involved creating a reduced version of the df_results_1k_plus_households DataFrame:

[16]:
```
df_regression_test = df_results_1k_plus_households.copy()
df_regression_test.
 ↪dropna(subset=['Proportion_of_children_in_female_householder_families_below_poverty_level',
df_regression_test =␣
 ↪df_regression_test[['Zip','Proportion_of_children_in_female_householder_families_below_pove
 ↪copy()
```

[17]:
```
df_regression_test
```

[17]:
```
          Zip  \
81      99505
94      66027
128     42223
133     28547
141     23665
…         …
31013   35203
31018   37403
31019   45203
31020   85351
31021   85622

       Proportion_of_children_in_female_householder_families_below_poverty_level
\
81                                                    0.469136
94                                                    0.781250
128                                                   0.554813
133                                                   0.707692
141                                                   0.580247
…                                                        …
31013                                                 0.505376
31018                                                 0.240310
31019                                                 0.841880
31020                                                 1.000000
31021                                                 0.000000

       Proportion_of_children_in_married_couple_families_below_poverty_level
81                                                    0.014045
94                                                    0.058471
128                                                   0.088429
133                                                   0.023663
```

```
141                                                 0.020168
…                                                        …
31013                                               0.000000
31018                                               1.000000
31019                                               0.000000
31020                                               0.000000
31021                                               0.000000

[16908 rows x 3 columns]
```

I then converted the two different variable columns into two different rows for each zip code using pd.melt(), which would make it easier to create categorical or 'dummy' variables for the regression analysis:

[18]:
```
df_regression_test_melt = pd.melt(df_regression_test.copy(), id_vars = ['Zip'])␣
↪# https://pandas.pydata.org/pandas-docs/version/0.20/generated/pandas.melt.
↪html
df_regression_test_melt
```

[18]:

	Zip	variable	value
0	99505	Proportion_of_children_in_female_householder_f…	0.469136
1	66027	Proportion_of_children_in_female_householder_f…	0.781250
2	42223	Proportion_of_children_in_female_householder_f…	0.554813
3	28547	Proportion_of_children_in_female_householder_f…	0.707692
4	23665	Proportion_of_children_in_female_householder_f…	0.580247
…	…	…	…
33811	35203	Proportion_of_children_in_married_couple_famil…	0.000000
33812	37403	Proportion_of_children_in_married_couple_famil…	1.000000
33813	45203	Proportion_of_children_in_married_couple_famil…	0.000000
33814	85351	Proportion_of_children_in_married_couple_famil…	0.000000
33815	85622	Proportion_of_children_in_married_couple_famil…	0.000000

[33816 rows x 3 columns]

The following code block uses pd.get_dummies to generate categorical variables, then renames the resulting column for better legibility.

[19]:
```
df_regression_test_melt = pd.get_dummies(data = df_regression_test_melt.copy(),␣
↪columns=['variable'], drop_first=True)
df_regression_test_melt.
↪rename(columns={'variable_Proportion_of_children_in_married_couple_families_below_poverty_l
↪'in_married_household','value':
↪'proportion_below_poverty_level'},inplace=True)
df_regression_test_melt
```

[19]:

	Zip	proportion_below_poverty_level	in_married_household
0	99505	0.469136	0
1	66027	0.781250	0

```
2       42223                          0.554813                      0
3       28547                          0.707692                      0
4       23665                          0.580247                      0
…       …                               …                            …
33811   35203                          0.000000                      1
33812   37403                          1.000000                      1
33813   45203                          0.000000                      1
33814   85351                          0.000000                      1
33815   85622                          0.000000                      1

[33816 rows x 3 columns]
```

With this table in place, I was able to perform the regression analysis.

[20]:
```
y = df_regression_test_melt['proportion_below_poverty_level'] # Contains the␣
 ↪list of scores for the current grade (or for the school total in the case of␣
 ↪the 'Total' column)
x_vars = df_regression_test_melt[['in_married_household']]
x_vars = sm.add_constant(x_vars)
model = sm.OLS(y,x_vars)
results = model.fit() # the resulst variable contains the information needed to␣
 ↪fill in the other rows within the DataFrame.
results.summary()
```

[20]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                                 OLS Regression Results
==============================================================================
==========
Dep. Variable:     proportion_below_poverty_level   R-squared:
0.423
Model:                                        OLS   Adj. R-squared:
0.423
Method:                             Least Squares   F-statistic:
2.476e+04
Date:                            Thu, 12 Aug 2021   Prob (F-statistic):
0.00
Time:                                    22:05:29   Log-Likelihood:
11261.
No. Observations:                           33816   AIC:
-2.252e+04
Df Residuals:                               33814   BIC:
-2.250e+04
Df Model:                                       1
Covariance Type:                        nonrobust
==============================================================================
========
```

```
                            coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
--------
const                      0.3945      0.001    295.734      0.000       0.392
0.397
in_married_household      -0.2968      0.002   -157.348      0.000      -0.301
-0.293
==============================================================================
Omnibus:                     1252.129   Durbin-Watson:                   1.823
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1842.596
Skew:                           0.365   Prob(JB):                         0.00
Kurtosis:                       3.880   Cond. No.                         2.62
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
"""
```

My second regression analysis aimed to evaluate the impact of family type (married vs. female-householder-only) and education level (no high school diploma; high school diploma/equivalent; associate's/some college; and bachelor's or higher) on poverty status. This first involved retrieving data on income for both family type and education.

[21]:
```
df_regression_test_2 = df_results_1k_plus_households.copy()
df_regression_test_2.
 ↪dropna(subset=['Proportion_of_married-couple_families_below_the_poverty_level_where_household
 ↪equivalent',␣
 ↪'Proportion_of_married-couple_families_below_the_poverty_level_where_householder\'s_highest
 ↪'Proportion_of_married-couple_families_below_the_poverty_level_where_householder\'s_highest
 ↪'Proportion_of_female-householder_families_below_the_poverty_level_where_householder_did_no
 ↪'Proportion_of_female-householder_families_below_the_poverty_level_where_householder\'s_hig
 ↪equivalent',␣
 ↪'Proportion_of_female-householder_families_below_the_poverty_level_where_householder\'s_hig
 ↪'Proportion_of_female-householder_families_below_the_poverty_level_where_householder\'s_hig
df_regression_test_2 =␣
 ↪df_regression_test_2[['Zip','Proportion_of_married-couple_families_below_the_poverty_level_
 ↪equivalent',␣
 ↪'Proportion_of_married-couple_families_below_the_poverty_level_where_householder\'s_highest
 ↪'Proportion_of_married-couple_families_below_the_poverty_level_where_householder\'s_highest
 ↪'Proportion_of_female-householder_families_below_the_poverty_level_where_householder_did_no
 ↪'Proportion_of_female-householder_families_below_the_poverty_level_where_householder\'s_hig
 ↪equivalent',␣
 ↪'Proportion_of_female-householder_families_below_the_poverty_level_where_householder\'s_hig
 ↪'Proportion_of_female-householder_families_below_the_poverty_level_where_householder\'s_hig
 ↪copy()
```

[22]:
```
df_regression_test_2
```

[22]:
```
          Zip  \
81      99505
128     42223
146     80902
159     31905
161     84005
…         …
30977   85375
30985   00925
31004   43215
31020   85351
32067   92637

       Proportion_of_married-couple_families_below_the_poverty_level_where_house
holder_did_not_graduate_high_school  \
81                                                    0.000000
128                                                   0.101449
146                                                   0.000000
159                                                   0.000000
161                                                   0.000000
…                                                        …
30977                                                 0.017857
30985                                                 0.794872
31004                                                 0.000000
31020                                                 0.042735
32067                                                 0.000000

       Proportion_of_married-couple_families_below_the_poverty_level_where_house
holder's_highest_education_=_high_school_graduate/equivalent  \
81                                                    0.041667
128                                                   0.148541
146                                                   0.177778
159                                                   0.180897
161                                                   0.039882
…                                                        …
30977                                                 0.092789
30985                                                 0.763158
31004                                                 0.157895
31020                                                 0.050662
32067                                                 0.115234

       Proportion_of_married-couple_families_below_the_poverty_level_where_house
holder's_highest_education_=_some_college_or_associate's_degree  \
81                                                    0.000000
128                                                   0.065651
```

```
146                                                 0.022084
159                                                 0.036606
161                                                 0.036001
…                                                        …
30977                                               0.020644
30985                                               0.285714
31004                                               0.000000
31020                                               0.039694
32067                                               0.039238

       Proportion_of_married-couple_families_below_the_poverty_level_where_house
holder's_highest_education_level_=_bachelor's_degree_or_higher  \
81                                                  0.000000
128                                                 0.000000
146                                                 0.000000
159                                                 0.004343
161                                                 0.005251
…                                                        …
30977                                               0.022065
30985                                               0.413793
31004                                               0.011518
31020                                               0.033682
32067                                               0.051859

       Proportion_of_female-householder_families_below_the_poverty_level_where_h
ouseholder_did_not_graduate_high_school  \
81                                                  1.000000
128                                                 0.000000
146                                                 1.000000
159                                                 0.000000
161                                                 0.828571
…                                                        …
30977                                               0.000000
30985                                               0.740331
31004                                               0.000000
31020                                               0.000000
32067                                               0.409091

       Proportion_of_female-householder_families_below_the_poverty_level_where_h
ouseholder's_highest_education_=_high_school_graduate/equivalent  \
81                                                  1.000000
128                                                 0.906667
146                                                 0.200000
159                                                 0.379310
161                                                 0.420635
…                                                        …
30977                                               0.084656
```

```
30985                                                 0.444444
31004                                                 0.577778
31020                                                 0.061947
32067                                                 0.097826

       Proportion_of_female-householder_families_below_the_poverty_level_where_h
ouseholder's_highest_education_=_some_college_or_associate's_degree  \
81                                                    0.000000
128                                                   0.503145
146                                                   0.900000
159                                                   0.468085
161                                                   0.424354
…                                                          …
30977                                                 0.000000
30985                                                 0.642202
31004                                                 1.000000
31020                                                 0.050725
32067                                                 0.140940

       Proportion_of_female-householder_families_below_the_poverty_level_where_h
ouseholder's_highest_education_level_=_bachelor's_degree_or_higher
81                                                    0.000000
128                                                   0.505051
146                                                   0.883721
159                                                   0.386364
161                                                   0.057325
…                                                          …
30977                                                 0.254386
30985                                                 0.655172
31004                                                 0.421687
31020                                                 0.000000
32067                                                 0.072917

[13919 rows x 9 columns]
```

Next, I once again 'melted' various columns into the same column in order to facilitate the creation of categorical variables. I also created columns that would store these categorical variables.

[23]:
```
df_regression_test_2_melt = pd.melt(df_regression_test_2.copy(), id_vars =␣
↪['Zip'])
df_regression_test_2_melt['Married'] = 0
df_regression_test_2_melt['highest_ed_=_high_school_grad'] = 0
df_regression_test_2_melt['highest_ed_=_some_college_or_associate\'s'] = 0
df_regression_test_2_melt['highest_ed_=_bachelor\'s_or_higher'] = 0
```

[24]:
```
df_regression_test_2_melt
```

[24]:

	Zip	variable	value	Married	highest_ed_=_high_school_grad	highest_ed_=_some_college_or_associate's	highest_ed_=_bachelor's_or_higher
0	99505	Proportion_of_married-couple_families_below_th…	0.000000	0	0	0	0
1	42223	Proportion_of_married-couple_families_below_th…	0.101449	0	0	0	0
2	80902	Proportion_of_married-couple_families_below_th…	0.000000	0	0	0	0
3	31905	Proportion_of_married-couple_families_below_th…	0.000000	0	0	0	0
4	84005	Proportion_of_married-couple_families_below_th…	0.000000	0	0	0	0
…	…	…	…	…	…	…	…
111347	85375	Proportion_of_female-householder_families_belo…	0.254386	0	0	0	0
111348	00925	Proportion_of_female-householder_families_belo…	0.655172	0	0	0	0
111349	43215	Proportion_of_female-householder_families_belo…	0.421687	0	0	0	0
111350	85351	Proportion_of_female-householder_families_belo…	0.000000	0	0	0	0
111351	92637	Proportion_of_female-householder_families_belo…	0.072917	0	0	0	0

```
111348                                          0
111349                                          0
111350                                          0
111351                                          0

[111352 rows x 7 columns]
```

The output of the following for loop served as a reference for which column numbers corresponded to which variables.

[25]:
```
for i in range(len(df_regression_test_2_melt.columns)):
    print("Column",i,":\t",df_regression_test_2_melt.columns[i])
```

```
Column 0 :       Zip
Column 1 :       variable
Column 2 :       value
Column 3 :       Married
Column 4 :       highest_ed_=_high_school_grad
Column 5 :       highest_ed_=_some_college_or_associate's
Column 6 :       highest_ed_=_bachelor's_or_higher
```

In the next for loop, I filled in the categorical variables by seeing whether certain keywords ('married', 'some_college', etc.) were present in the variable column. For instance, given the variable 'Proportion_of_married-couple_families_below_the_poverty_level_where_householder_did_not_graduate_high_school', the for loop returned 1 for the 'Married' column and 0 for the other columns.

[26]:
```
for i in range(len(df_regression_test_2_melt)):
    variable = df_regression_test_2_melt.iloc[i, 1]
    if 'married' in variable:
        df_regression_test_2_melt.iloc[i, 3] = 1
    if 'high_school_graduate' in variable:
        df_regression_test_2_melt.iloc[i, 4] = 1
    if 'some_college' in variable:
        df_regression_test_2_melt.iloc[i, 5] = 1
    if 'bachelor' in variable:
        df_regression_test_2_melt.iloc[i, 6] = 1
```

[27]:
```
df_regression_test_2_melt.iloc[0,1]
```

[27]:
```
'Proportion_of_married-couple_families_below_the_poverty_level_where_householder
_did_not_graduate_high_school'
```

[28]:
```
df_regression_test_2_melt.rename(columns={'value':
 ↪'proportion_below_poverty_level'},inplace=True)
df_regression_test_2.to_csv('marriage_education_poverty_regression.csv')
df_regression_test_2_melt
```

[28]:

	Zip	variable	\
0	99505	Proportion_of_married-couple_families_below_th…	
1	42223	Proportion_of_married-couple_families_below_th…	
2	80902	Proportion_of_married-couple_families_below_th…	
3	31905	Proportion_of_married-couple_families_below_th…	
4	84005	Proportion_of_married-couple_families_below_th…	
…	…	…	
111347	85375	Proportion_of_female-householder_families_belo…	
111348	00925	Proportion_of_female-householder_families_belo…	
111349	43215	Proportion_of_female-householder_families_belo…	
111350	85351	Proportion_of_female-householder_families_belo…	
111351	92637	Proportion_of_female-householder_families_belo…	

	proportion_below_poverty_level	Married	\
0	0.000000	1	
1	0.101449	1	
2	0.000000	1	
3	0.000000	1	
4	0.000000	1	
…	…	…	
111347	0.254386	0	
111348	0.655172	0	
111349	0.421687	0	
111350	0.000000	0	
111351	0.072917	0	

	highest_ed_=_high_school_grad	\
0	0	
1	0	
2	0	
3	0	
4	0	
…	…	
111347	0	
111348	0	
111349	0	
111350	0	
111351	0	

	highest_ed_=_some_college_or_associate's	\
0	0	
1	0	
2	0	
3	0	
4	0	
…	…	
111347	0	

```
111348                                          0
111349                                          0
111350                                          0
111351                                          0

        highest_ed_=_bachelor's_or_higher
0                                       0
1                                       0
2                                       0
3                                       0
4                                       0
…                                       …
111347                                  1
111348                                  1
111349                                  1
111350                                  1
111351                                  1

[111352 rows x 7 columns]
```

With the table complete, I performed a regression that used proportion_below_poverty_level as the dependent variable and various family type/education level values as the independent variables.

[29]:
```
y = df_regression_test_2_melt['proportion_below_poverty_level']
x_vars = df_regression_test_2_melt[['Married',
       'highest_ed_=_high_school_grad',
       'highest_ed_=_some_college_or_associate\'s',
       'highest_ed_=_bachelor\'s_or_higher']]
x_vars = sm.add_constant(x_vars)
model = sm.OLS(y,x_vars)
results_2 = model.fit()
results_2.summary()
```

[29]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                                OLS Regression Results
==============================================================================
==========
Dep. Variable:     proportion_below_poverty_level   R-squared:
0.338
Model:                                        OLS   Adj. R-squared:
0.337
Method:                             Least Squares   F-statistic:
1.418e+04
Date:                            Thu, 12 Aug 2021   Prob (F-statistic):
0.00
Time:                                    22:06:06   Log-Likelihood:
37256.
```

```
No. Observations:                          111352   AIC:
-7.450e+04
Df Residuals:                              111347   BIC:
-7.445e+04
Df Model:                                       4
Covariance Type:                        nonrobust
==============================================================================
============================
                                                coef    std err          t
P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
----------------------------
const                                         0.3750      0.001    323.203
0.000       0.373       0.377
Married                                      -0.1957      0.001   -188.537
0.000      -0.198      -0.194
highest_ed_=_high_school_grad                -0.0886      0.001    -60.346
0.000      -0.091      -0.086
highest_ed_=_some_college_or_associate's     -0.1207      0.001    -82.230
0.000      -0.124      -0.118
highest_ed_=_bachelor's_or_higher            -0.2112      0.001   -143.887
0.000      -0.214      -0.208
==============================================================================
Omnibus:                    19326.187   Durbin-Watson:                   1.860
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            50054.262
Skew:                           0.958   Prob(JB):                         0.00
Kurtosis:                       5.668   Cond. No.                         5.39
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
"""
```

[30]:
```
end_time = time.time()
run_time = end_time - start_time
run_minutes = run_time // 60
run_seconds = run_time % 60
print("Completed run at",time.ctime(end_time),"(local time)")
print("Total run time:",'{:.2f}'.format(run_time),"second(s)␣
↪("+str(run_minutes),"minute(s) and",'{:.2f}'.
↪format(run_seconds),"second(s))") # Only valid when the program is run␣
↪nonstop from start to finish
```

```
Completed run at Thu Aug 12 22:06:06 2021 (local time)
Total run time: 123.55 second(s) (2.0 minute(s) and 3.55 second(s))
```