Introduction

One goal of data analysis is to describe the world around us. Businesses, governments, and economic researchers have a need to know what is happening in the world and what trends are developing. This information is necessary for setting prices on products, paying employees competative salaries, demographics of voters, etc. One of the largest data sets in the United States is the US Census data. By analyzing this large data set, one can gleam a picture of the world around us and the data points concerning each individual in the United States.

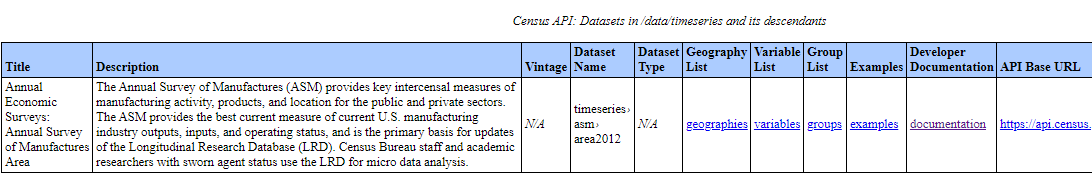
This project provides an example of how US Census data may be leveredged using Python programing to identify trends using inferential statistical testing. Specifically, the example looks at salary and employment data across differing characteristics of sex, race, age, and geographical. This is a common task of macro economists and one that should be straight forward for a professional data scientist.

Data Description

One of the largest data sets in the United States is the US Census data which polls each citizen on their race, sex, employment status, salary, family composition, business inventories, mortality/birth rates, population growth, and a host of other characteristics. By analyzing this data, a data scientist may be able to create a picture of what the demographic makeup is in the US. Furthermore, the data scientist may use the data to perform statistical analysis to identify underlying correlations which may be prevelant in society.

The United States Census Bureau states on their website that their mission is to “…serve as the nation’s leading provider of quality data about its people and economy”( <https://www.census.gov/about.html>). It is a Federal requirement for each citizen in the United States to fill out the Census form and submit it to the Census Bureau. Naturally, the Bureau has a nearly complete population of all citizen data that no other data provider is likely to have.

The data itself is made up of a collection of tables which report different characteristics of the total data set. For this project I focused on time series data in order to be able to look at data sets and compare them across time. A listing of the time series data tables can be found at <https://api.census.gov/data/timeseries.html>. The website gives a title and description of each table, followed by a Dataset Name, the relevant Geography, Group, and Variable lists. Each table also has an example of how the data may be retrieved, and a URL link to fascilitate the API’s from which data users may retrieve the Census data.

Example table entry for the timeseries data list:

For this project I referenced the table titled “Time Series Longitudinal Employer-Household Dynamics QWI: sex by age”. This was selected to look at sex, age, and race on a quarterly basis and compare these qualitative factors against the employment status and total salary across different groups.

The data is made accessible using API’s which reference HTML data via the Census Bureau website. A python library called census 0.5 (<https://pypi.org/project/census/0.5/>) is available which can be used to access the Census data either using distinct Library commands or through referencing a website link and retrieving the data via web address. I used the later approach as I found it provided more flexibility after importing the data set. Also, by referencing the data directly using the census requests.get function, I could access data without having to use an API key. A key is required when using the census library code indirectly rather than by referencing the website data directly.

Example of code used to call the primary data set:

work\_age = requests.get("https://api.census.gov/data/timeseries/qwi/sa?get=sex,Emp,agegrp,EarnS,race,education&for=metropolitan**%20s**tatistical**%20a**rea/micropolitan**%20s**tatistical**%20a**rea:\*&in=state:01&time=2019")

Data was then imported to a pandas dataframe as a json file as follows:

df\_work\_age1 = pd.DataFrame(work\_age.json())

The table API requests.get function only allowed me to reference one state’s data at a time so a while loop was used to append all of the state data results into one pandas dataframe. Data cleanup was then necessary to assign header names, data types for columns, and remove blank or NaN data entries.

The data set contained codes for nearly all fields and I was not able to reference the ancilary code deffinition tables the same way that the census data was retrieved. I found that the code lists were provided via pdf (example: <https://www2.census.gov/programs-surveys/acs/tech_docs/code_lists/2019_ACS_Code_Lists.pdf>) on the census website so I recreated reference tbales in excel and uploaded them as additional dataframes in my code. By having the code tables added as their own distinct dataframes, I was able to join the code meanings next to the code fields in the census data. This joining of code with meaning was necessary to produce and decifer the statistical results since the data analysis looking at only the codes appeared nonsensical.

Methodology

Inferential statistical testing was performed to observe correlations between demographic characteristics and their associated salary and employment data. A statistical test was performed on qualitative demographical catigories of sex, age, and race. The salary was designated as the dependent variable and the qualitative category as the indipendent variable. Dummy variables were assigned to each qualitative data chategory; for example, 1 for male, 2 for female. An Ordinary Least Squares was then performed using the ols() function in the following library imports:

*#Import stats libraries for ols*

**from** **statsmodels.formula.api** **import** ols

**import** **statsmodels.api** **as** **sm**

**from** **scipy** **import** stats

To display the data, the matplotlib.pyplot and seaborn libraries were imported. Histograms were used to display the count per salary group for each qualitative characteristic of sex, race, and age.

Results

Discussion

Conclusion