**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.

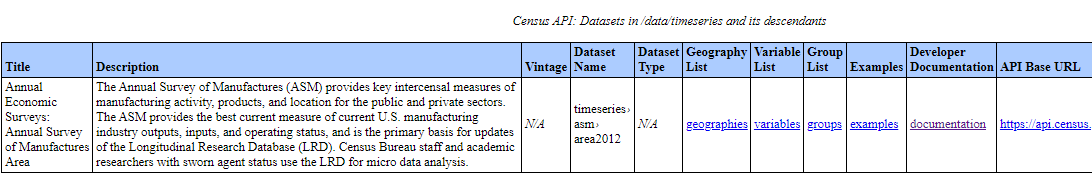
This project provides an example of how US Census data may be leveredged using Python programing to identify trends using inferential statistical testing. Specifically, the example looked at gender of US citizens and whether it could indicate an approximate impact to monthly salary figures.

**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 total salary per gender.

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 tables 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 decipher 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. An ordinary least squares linear regression was performed with a statistical test performed on qualitative demographical catigories of gender. 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, 0 for male, 1 for female. The total sample size was 20,768 which represents total locations surveyed by the UC census; each of the 20,768 locations sampled represents a total population salary figures for that location.

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

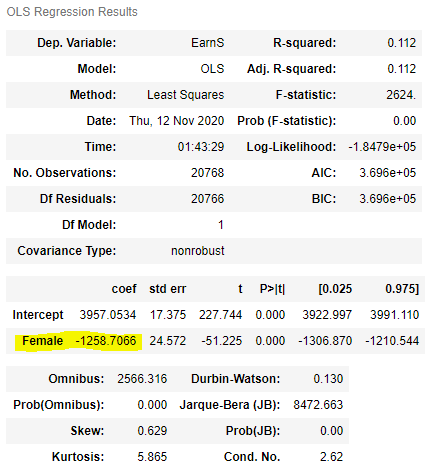
To display the data, the matplotlib.pyplot and seaborn libraries were imported. Histograms were used to display the count per salary group the gender qualitative characteristic.

**Results**

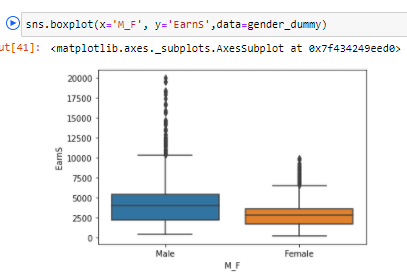
The Ordinary Least Squares linear regression testing showed a clear negative coefficient factor associated with being Female when modeling for salary amount. The coefficient value represents the mean change in the dependent variable of salary given a one unit change in the independent variable of male or female. In this instance the dependent variable was assigned using a dummy variable of 1 or 0 for female or male (if female then 1). Since female would be assigned a one, this indicates a one-unit change in the independent variable; therefore, the average change in salary as a result of being female is the amount of the coefficient. In the statistical testing, the coefficient was -1,258.71. In otherwords, females in the united states receives on average -$1,258.71 less salary per month.

Caveates to the finding that women receive -$1,258.71 less salary per month on average could be a host of reasons such as differing amounts of females employed vs males. Also, males may choose to hold different jobs on average which pay different salaries.

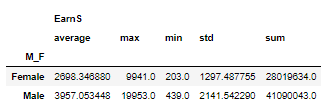
The Gender proved to not be a statistically significant factor for predicting salary using a ordinary least squares method for estimating unknown parameters in a linear regression model. Statistical testing indicated that the P statistic was less than the t test value. This does not invalidate the afformentioned coefficient figure and merely indicates that there are many other factors necessary for predicting the average salary per location surveyed.



The findings from the OLS testing were supported by pivoting the total data set and looking at the mean, min, max, sum of the salary of women compared with men. More shocking than the average decrease women received was that the average max per location earnings by women was significantly less than men and appeared to be capped well below the average men receive.

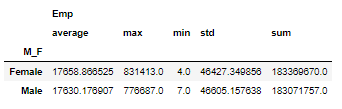


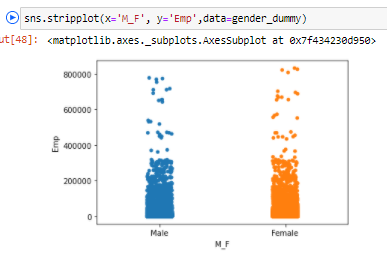
EarnS = Average salary per location per month:

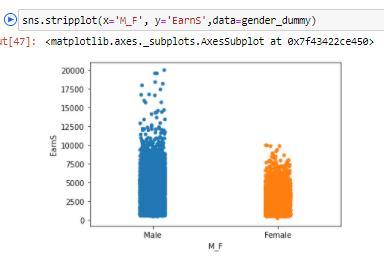


From the boxplot and pivot table above, we see that the mean salary for men is higher than women at $3,957 for men vs $2698 for women. The boxplot shows that there is a large amount of high end earning location which are outliers from the normal population; however, men appear to have significantly more and higher earners. Surprisingly, the max earnings avg per location per month is $19,953 for men and $9,941 for women.

Emp = total amount of people employed:







The pivot table of total citizens employed (“Emp” variable) shows that there were in fact more females surveyed despite the lower salary figures. The stripplot graphs above show that despite there being comparable populations of both men and women surveyed, the male average salaries per location far exceeded $10,000 per month for many locations while average salary per location for women was capped below $10,000.

**Discussion**

This project served to display the ability to analyze US Census data using python and draw meaningful conclusions from a very large, open source data set. The US Census data is accessible for free to anyone who is capable of accessing it. Unfortunately, the GUI provided is not as flexible as Python for performing analysis and does not lend itself for creating statistical testing. By using Python, a data scientist would be able to approach many similar questions regarding the makeup and trends behind the US population’s data. By performing exercises like this, data scientists can provide a clear picture of our society and help to draw attention to data inconsistencies which have consequence in our daily lives.

**Conclusion**

In this project the data showed that women on average per location make less per month than their mail counterparts. This example is limited in scope and does not seek to answer why that is the case, what locations have the largest discrepancies, or what other factors are contributing to differences in pay. Additional research would be necessary to provide conclusive findings; however, this project shows both the ease of accessing data using the US Census resource and the flexibility provided by Python in retrieving the data, preparing it for analysis and fascilitating analysis and meaningful findings.