**Mini Project 2**

**IST 652 Scripting for Data Analysis**

**Paul Sujith Rayi**

**Data Source:**

My dataset is from Data Catalog of the U.S. government. The U.S. government releases datasets to public for data analysis and other educational uses. This, I think is a wonderful thing that the U.S. government does to show openness to its public and to students studying at various levels in the country. The datasets range from topics such as Agriculture, Climate, Consumer, Ecosystems etc. Here is the link to the data catalog website.

<https://catalog.data.gov/dataset>

My particular dataset is a JSON dataset that I extracted from the same data catalog website. It is a demographic census dataset of New York City (NYC). It consists of all zip codes in NYC and people living in those zip codes with regards to their sex, race, citizenship status and public assistance.

I downloaded the JSON dataset which I am attaching with this report. But in my python file, I have extracted the JSON directly from the website using urllib.request package. Here’s the link to the website from which I directly extracted the JSON data.

<https://data.cityofnewyork.us/api/views/kku6-nxdu/rows.json?accessType=DOWNLOAD>

**Data Analysis:**

After extracting the JSON dataset, I parsed and decoded it using the read and decode function. Then I printed the first 500 words/instances of my dataset. Then I used the json.loads function to convert my dataset into a better structured format. The json.loads function converts your JSON dataset into dictionaries and lists. Then I looked at the keys of my structured dataset of which the outermost data structure is a dictionary. I printed the keys on the console. Then I selected one key ‘meta’ and printed out the content it contains on the console. It is a mixture of dictionaries and lists. Similarly, I did the same for my ‘data’ key and it was a list of lists.

I looked at the metadata in the website for a minute and was able to recognize that each list in my ‘data’ key corresponds to each row in my JSON dataset. Hence, I converted the list of lists into a dataframe so that each list represents a row in the dataframe. I then removed column index numbers from 0 – 7, as we will be not needing them for our analysis. Then I assigned column names to the remaining columns. I got to know the column names by looking at the metadata in the website. Then I removed all columns starting with percent in their names, as I thought we will be not needing them for our analysis. Similarly, I removed all my columns ending with the name ‘total’ as I thought we will not be needing them too for our analysis. Then I checked to see if any of my remaining columns contained any missing values. It turned out that none of my columns contained any missing values. This is a positive sign for us as it shows that our data is clean with respect to missing values.

After looking at the dataset now, I was able to figure out that there were four broad categories of demographic census data in the zip codes of NYC. The categories are sex, race, citizenship status and public assistance. After figuring out this I thought to subdivide my dataset into separate datasets based on these categories and do analysis.

Hence my first subdivided dataframe consists of all the columns related to sex and also the column pertaining to the zip codes. I ran summary statistics and converted some string columns to numeric which is apt for our analysis. I then changed the column names for all my columns in this dataframe and the new column names did not contain any spaces. Having spaces in column names is not a good pythonic practice. I then checked for unique values in my gender unknown column. There was only one unique value and that was zero. This basically tells us that in all the zip codes in NYC, there are either males or females and none of people whose gender is not known. Hence, I decided to delete the column and deleted it. After that I used the query function to extract all the zip codes where the population of females is greater than the population of males. After that I did a visualization of the same with my x-axis representing the zip codes and my y-axis representing the count of females and count of males for those zip codes. I then output the extracted dataframe into an excel file that gets saved in the same directory as the one you are running the python script. I named my excel file to be output 1. Here’s the summary of it.

**Output 1:** The column names is the header row. There are three columns in this file. The first column consists of all the zip codes where the count of female is greater than the count of males. There are a total of 60 rows which tell us that there are more females in these sixty zip codes as compared to males.

Now I again used the query function to extract the zip codes where the count of females is greater than 10.0 which is the mean count of females in all zip codes. I wanted to see all the zip codes where women where greater than their average number. I then visualized this dataframe with my x-axis being the zip codes and my y-axis being the count of females. And by the way, the dataframe returned also contains the values for count of males too in these zip codes. This is how the query function works. I then output my extracted dataframe as an excel file with my header row being the column names. I named my excel file output 2. Here’s the summary of it.

**Output 2:** There is a header row consisting of our three columns namely zip codes, count of females and count of males. The columns consist of the respective values of zip codes, count of females and count of males. There are a total of 48 rows excluding the header row, and we can say that for these 48 zip codes the count of females is more than the mean count of females in the city.

Now I subdivided my dataframe based on race. I did some summary statistics using the describe function. I then converted all my columns except for zip codes column into numeric. I had to do this as the values of the columns are counts and it made more analytical sense to convert them into integers, rather to keep them as strings. I again ran summary statistics over them using the describe function. I then changed the columns names for all my columns in this dataframe and the new column names did not contain any spaces. Having spaces in column names is not a good pythonic practice. I then checked for unique values in two of my columns. After that, I ran the query function twice to extract all the zip codes where count of hispanics and blacks is greater than 20.0. I then visualized this dataframe using the plot function where my x-axis consists of the zip codes and my y-axis consists of count of hispanics and blacks. After this, I output my dataframe into an excel file with the header row being the names of columns and I named my output file output3. Here’s the summary of my output.

**Output 3:** The values in my zip codes column are all zip codes where the count of blacks and hispanics in NYC is greater than 20.0. The rest of my columns values are counts for the respective races in those zip codes. After looking at the output file. I can say that there are only three zip codes in NYC where the count of hispanics and blacks is greater than 20.0.

I then queried my subdivided dataframe to extract all zip codes where count of other ethnicity is greater than 5.0 and the count of ethnicity unknown is less than 4.0. I then visualized the extracted dataframe using the plot function where my x-axis is the zip codes and my y-axis consists of the count of other ethnicity and ethnicity unknown in those zip codes. I output my extracted dataframe into an excel file and named my excel file output4. Here’s the summary of it.

**Output 4:** The header row is the names of all my columns in the extracted dataframe. The first column consists of all the zip codes that matched my query criteria. The rest of the columns are all counts for ethnicities in those zip codes. Looking at the output file I can say that there a total of seven zip codes where other ethnicity is greater than 5.0. and ethnicity unknown is less than 4.0.

I now subdivided my dataframe based on citizenship status. I did some summary statistics on my subdivided dataframe. After that I converted all column types from string to integer except for my zip codes column. I then performed summary statistics again. I then changed my column names by giving them names with no spaces. Column names with spaces in them is not a good pythonic practice. I then checked for unique values in one of my columns which is citizen status unknown and found out that it contains only one value throughout and that is zero. Hence, I decided to delete that column as it is not required for further analysis and based on that value, I can say that there are no people in any NYC zip codes who do not know what their citizenship status is. I then queried my subdivided dataframe to extract all zip codes in NYC where the number of US citizens is greater than 200.0. I then visualized my extracted dataframe using the plot function where my x-axis is zip codes and my y-axis is count of us citizens. I then output my extracted dataframe into an excel file and named the file output5. Here’s the summary of the file.

**Output 5:** The header row is the column names. The first column consists of all zip codes where us citizens number is greater than 200.0. The other columns consist of the counts of people with different citizenship statuses in those zip codes. There are a total of five rows in the output file excluding the header row which tells us that in these five zip codes the count of us citizens is greater than 200.0.

I now queried my subdivided dataframe and extracted zip codes where the count of permanent residents is greater than 3.0 and the count of other citizen status is less than 2.0. I visualized my extracted dataframe using the plot function where my x-axis consists of zip codes and my y-axis consists of the count of permanent residents and other citizens. I then output my extracted dataframe into an excel file and named it output6. Here’s the summary of it.

**Output 6:** The header row is the column names. The first column consists of all zip codes where us permanent residents number is greater than 3.0 and other citizen status is less than 2.0. The other columns consist of the counts of people with different citizenship statuses in those zip codes. There are a total of four rows in the excel file excluding the header row. This tell us that in these zip codes, US permanent residents are more than 3.0 and other citizens are less than 2.0.

I now subdivided my dataframe based on columns with regards to public assistance. I then did summary statistics on my subdivided data frame. I then converted all columns except zip codes to numeric as they contained values of counts which are better analyzed as integers. After that I again did summary statistics on my subdivided dataframe. I then renamed my column names by removing the spaces in them. It is not a good pythonic practice to have spaces in column names. I then checked for unique values in one of my columns which is count\_public\_assistance\_unknown. After running the unique function on that column, it emerged that there is only one unique value and that is zero. This tells us that there is no zip code in NYC where people do not know their public assistance status live. Hence, I deleted this column from my subdivided dataframe. I then queried my subdivided dataframe to extract all the zip codes where the number of people who receive public assistance is less than 50.0 and the number of people who do not receive public assistance is greater than 100.0. I then did visualization on my extracted dataframe using the plot function where my x-axis is the zip codes and my y-axis is the count of people who receive and do not receive public assistance in those zip codes. I then output my extracted dataframe into an excel file and named it output7. Here’s the summary of it.

**Output 7:** The header row is the column names. The first column consists of all zip codes where people who receive public assistance is less than 50.0 and people who do not receive public assistance is greater than 100.0. The other two columns are the counts of people who receive and do not receive public assistance. There are a total of two rows excluding the header row in this output file. This tells us that in these zip codes the number of people who do not receive public assistance is way higher than the ones who receive public assistance. Based on this output, I can say that if NYC officials want to check whether public assistance is evenly distributed or not in the city, then they can look at these zip codes and get some insight. This recommendation though is based on just looking at the numbers and is not a complete picture. Some other factors that help us get the complete picture are median income of households in these zip codes, ethnicities of households in these zip codes, citizenship status etc.

I now extracted a dataframe consisting of all the zip codes where the count of people who receive public assistance is greater than 100.0 and who do not receive public assistance is less than 150.0. I then visualized my extracted dataframe using the plot function where my x-axis is the zip codes and my y-axis consists of people who receive and do not receive public assistance. I then output my extracted dataframe into an excel file and named it output8. Here’s the summary of the excel file.

**Output 8:** The header row consists of the names of the columns. The first column is the zip codes where people who receive public assistance are greater than 100.0 and people who do not receive public assistance are less than 150.0. The other two columns are the number of people who receive and do not receive public assistance in those zip codes. There are a total of two rows excluding the header row. This tells us that for these two zip codes the people who receive public assistance is greater than 100.0 and people who do not receive public assistance is less than 150.0.

This is the end of my analysis of this demographic data of NYC zip codes. This is an exploratory analysis which shows us how to extract unstructured data from a website and then clean it and bring it to a form of structured data format on which analysis can be done. Thank you.