Neel Patel

My portion of the project focuses on poverty, cancer, and income rates and totals. Our group wanted to investigate the social determinants of health and health outcomes. To do so, we all investigated separate determinants and pooled the information together for analysis. As stated, I focused on poverty, cancer, and income, and to express this analysis, I created maps, graphs, and charts. All the data was taken from credible, open-source databases. Specifically, I used government websites with relatively modern data that were similar to the time they were recorded. The data for income and poverty was taken from the US Census Bureau, and the data for cancer was taken from the State Cancer Profile. The State Cancer Profile is a government website that uses data from the National Cancer Institute and the CDC. Population data for cancer percentages is incorporated from the ARHQ government database and then truncated into the excel sheet from the State Cancer Profile.

For importing the data into my code, I implemented the package, openpyxl. To understand the uses of this package, I conducted research from Oluwabukunmi Ige who helped me understand how to reference the data in dictionaries and lists. In the program, it is very simple to run this part as you simply hit the run button and all the data will file itself into the corresponding lists. When separating the data, I imported the data into NumPy arrays and regular lists, therefore I had to import NumPy into my code.

When creating the graphs, I created a class that would graph three separate independent variables with three separate dependent variables, ultimately creating nine different combinations of graphs. This class was called graph and there were two main methods defined within this class. One was called “code”, which returned a string letter to the user. This method compares the user inputs to two of the six different NumPy arrays that were used in the code. Within this method, I had to use the .sum() function when comparing lists since I got the following error: “The truth value of an array with more than one element is ambiguous. Use a.any() or a.all()”. The purpose of the .sum() function verifies that the user input matches the built-in functions that are present in the class. The next method was called “plot” and it was dependent on the previous method to be returned. The user would call plot and input the specific letter returned from “code” as its parameter. This is necessary because it matches the graph with the specific titles and data that are going to be plotted. Here, the package of matplot was used. The title of the graph and the plot of the data were merged to form the graph. Within this method, I also used the scipy package. This package is a statistics-based package that allows me to determine certain components of the graph. It allows me to determine the r-squared value which tells me how much of the variance in the dependent variable can be accounted for by the independent variable. This is important because we need this value to show how strong the relationship between the independent and dependent variables is. The slope and the r^2 value are presented above the graph which allows the user to conduct a simple analysis.

The next part of the code involves mapping. I created three separate maps that presented the counties of Florida. One map showed the median income per county, the next map showed the poverty estimate by county, and the third and final map showed the average annual cancer count by county. To create the three maps, I used the plotly package, and I had to run the following code to allow the figure factory method to work: “conda install plotly conda install geopandas”. Some of the code for the map was taken from this site. In order to create a map of Florida and have it split by county, I had to retrieve the FIPS numbers from each of my data sets. To do so, I put the FIPS codes into a list at the beginning of the code and then I referenced them in the appropriate position when calling them. So, I adjusted the code to include the FIPS values from my datasets. The next step was to incorporate which data I would like to have reflected upon the map. To do so, I referenced the data lists that I created earlier. After this, I changed the “binning\_endpoints” variable to account for which data is being used. It changes the keys so that the data is more organized on the map. The rest of the code is taken from the plotly site and then reconstructed for my data.

The biggest issue I had with the construction of the maps was the depreciation warning. This warning told me that shapely, a package I had to install, was going out of date and it was no longer viable to use such methods. Although, when I ran the code from plotly, the code outputted the intended map correctly. There was no problem with the code, but I received a very long message and several warnings about the deprecation. To overcome this issue, I imported the warnings package and then imported the ShapelyDeprecationWarning. Then, I used a function from this package and used it to ignore the warnings. This was done because plotly recommended that I used a JSON file and recalibrated certain files to implement the newer technique. Unfortunately, doing the JSON file conversion and then using pandas and geopandas to program the code seemed a lot more tedious.

The last part of the code is the horizontal bar graph. To construct this, I used matplotlib and plotted two different datasets: the average cancer counts and the total number of people in poverty by county. To do so, I used plt.barh() and there were similarities between the poverty and cancer totals. Obviously, there could be several confounding variables present since the populations vary, but you can still create a correlation between the data. Although, when creating the graph of the percentages of people with cancer and the percentages of people in poverty, there is a negative correlation within the data.

My portion of the slides presentation was cancer, poverty, and income slides. I used pictures from my code to explain what each of the graphs meant and what its purpose was in reference to our whole project.

Based on the data, the graph shows that as the poverty percentage increases, the general number of cancer counts decreases. The main reason for this is that people in poverty will most likely choose not to report cancer cases. This is because they cannot afford the treatment and they want to avoid paying the medical bills. Thus, this is the most likely reason for the general negative trend between the percentage of poverty and cancer.

Ige, Oluwabukunmi. “How To Extract Information from Your Excel Sheet Using Python.” *Medium*, Analytics Vidhya, 29 Sept. 2019, <https://medium.com/analytics-vidhya/how-to-extract-information-from-your-excel-sheet-using-python-5f4f518aec49>.

“Plotly.” *USA County Choropleth Maps in Python*, https://plotly.com/python/county-choropleth/.

US Census Bureau. “SAIPE State and County Estimates for 2020.” *Census.gov*, 16 Dec. 2021, https://www.census.gov/data/datasets/2020/demo/saipe/2020-state-and-county.html.

“State Cancer Profiles.” *State Cancer Profiles - Incidence Rates Table*, https://www.statecancerprofiles.cancer.gov/incidencerates/index.php?stateFIPS=12&areatype=county&cancer=001&race=00&sex=0&age=001&stage=999&year=0&type=incd&sortVariableName=rate&sortOrder=default#results.

“Social Determinants of Health Database.” *AHRQ*, https://www.ahrq.gov/sdoh/data-analytics/sdoh-data.html.