Dalton Poppino

Dr. White

Tools for Data Analysis

May 3, 2022

Immigration Data: What I learned

I decided to look at immigration data from the U.S. Census for this project. I am interested in the destination of immigrants. To do that, I grabbed the data from January of every year from 2005-to 2019 from the Census. I then looked at each line of data provided by each person surveyed. I kept all the data where the parents were not from the United States. This way, I can lower the amount of data collected. Here is where I ran into my first learning moment of this project.

I was working through the data and trying to minimize the size of each data frame. For each of the 15 API calls, there were over 100,000 surveys. In my first attempt, I made everything a data frame and appended each data as a data frame to an existing data frame. Processing the data this way took almost four hours to pull the 15 instances of the API call. After trying to pull the data four times unsuccessfully, I thought there must be a quicker way to work with the data. After a bit of research, I discovered that data frames take approximately ten times longer to work with when compared to dictionaries. After I made the changes in my code to iterate over the dictionaries, my time to pull the code dropped to about ten minutes.

I knew from the beginning I wanted to make maps of some sort to represent the data on my dash. I retrieved the dictionary that was part of the Census data variable description. It was an easy copy and paste, so I used it as a start for my immigration home city. I was able to take the city and split the string over the “,” that was dividing the city and state in the given data. Then I thought it would be going through the motion of the Zippopotamus assignment and was going to be able to plug and play that assignment right into the work I was doing. This assumption was the start of learning moment number two.

I kept trying to replicate the work of assignment 13, and I just couldn’t get the same output, and I was getting aggravated because, in my mind, it was the same info. It should have just worked. I started looking at the provided dictionary and realized that there were many entries with multiple cities and extra information with the state data. I didn’t need any of this additional data, so I needed to break down what was given even further. I finally got everything sorted, and again the API wouldn’t pull. Finally, I just tried to copy and paste an entry from the data frame, and it still didn’t work. At this moment, I was fuming, so I took a break and came back to it the next day. It then came to me that there was white space that I wasn’t considering. This problem taught me two things. First, white space is the invisible devil. Secondly, taking a bit of time is good when working on coding because you can become hyper-focused on things that aren’t important, and coming back with a rested mind can do wonders. After I cleared the white space, the Zippopotamus worked, and I had the latitude and longitude data that I needed.

Now I needed to know where the parents had emigrated. I got the dictionary from the Census variable data on the website. I knew that I would want data from the mother’s home country and father’s home country separately. So, once I added the name of each country for the mother and father, I was able to start merging data frames to get the final data frame. I merged the data from the Census and Zippopotamus APIs and the data from the parent’s home country variable data. Now I just needed to get a quantifiable variable to create plots. I learned the groupby function to count all the instances of duplicates and then dropped the duplicated entries. I reported this as my final data frame as it had all the information needed to start building my dash.

Creating the charts that I wanted to use to represent the data was easy to make outside of a dash environment. Learning the dash component was quite tricky. I took one of the examples off the website and just started tinkering with the different aspects of the dashes provided. Then I started plugging in the info for the plots I wanted to make. I spent a lot of time on stack overflow, learning how to do the things I wanted, like only having unique entries alphabetized in the dropdown. I also wanted to set my dropdown to open to a value that I knew would provide new data so you could tell the data was used. Callbacks seemed intuitive once I became aware of how all the items were named, change this, update that, got it. I am sure there are more capabilities within the callback function, but they weren’t needed for my project. Overall dash was all new for me, but once I figured out how each part of the dash worked with each other, it just made sense.

If I were to go back and do anything different, I would have been more thoughtful with building the data frame. There could be endless amounts of plots generated, and it would be an insightful tool if you could select a specific number of variables and select which type of plot you wanted to develop. It would take a ton of code and probably building a website to be able to navigate it intuitively. I would also look at ways to add multiple variables for each dropdown menu. I learned how to have the dropdown be able to select various entries but didn’t learn how to handle those in the plotly functions within dash.